



CCCCCCCC VV VV DDDDDDDD RRRRRRRR I I I I VV VV EEEEEEEEEE RRRRRRRR  
CCCCCCCC VV VV DDDDDDDD RRRRRRRR I I I I VV VV EEEEEEEEEE RRRRRRRR  
CC VV VV DD DD RR RR I I VV VV EE RR RR  
CC VV VV DD DD RR RR I I VV VV EE RR RR  
CC VV VV DD DD RR RR I I VV VV EE RR RR  
CC VV VV DD DD RRRRRRRR I I VV VV EE RR RR  
CC VV VV DD DD RRRRRRRR I I VV VV EEEEEEEEEE RRRRRRRR  
CC VV VV DD DD RR RR I I VV VV EE RR RR  
CC VV VV DD DD RR RR I I VV VV EE RR RR  
CC VV VV DD DD RR RR I I VV VV EE RR RR  
CC VV VV DD DD RR RR I I VV VV EE RR RR  
CC VV VV DD DD RR RR I I VV VV EE RR RR  
CC VV VV DD DD RR RR I I VV VV EE RR RR  
CC VV VV DD DD RR RR I I VV VV EE RR RR  
CC VV VV DD DD RR RR I I VV VV EE RR RR  
CCCCCCCC VV VV DDDDDDDD RR RR I I I I VV VV EEEEEEEEEE RR RR  
CCCCCCCC VV VV DDDDDDDD RR RR I I I I VV VV EEEEEEEEEE RR RR  
.....  
.....

LL I I I I SSSSSSSS  
LL I I I I SSSSSSSS  
LL SS SS SS SS  
LLLLLLLLLL I I I I SSSSSSSS  
LLLLLLLLLL I I I I SSSSSSSS

(3)	77	EXTERNAL AND LOCAL DEFINITIONS
(4)	244	STANDARD TABLES
(7)	431	CONTROLLER INITIALIZATION ROUTINE
(8)	459	UNIT INITIALIZATION ROUTINE
(9)	503	DRIVER SPECIFIC SUBROUTINES
(10)	538	FDT ROUTINE - TEST TRANSFER BYTE COUNT ALIGNMENT
(11)	574	START I/O ROUTINE
(15)	1054	INTERRUPT SERVICE ROUTINE
(16)	1267	REGISTER DUMP ROUTINE

0000 1 .TITLE CVDRIVER - VAX/VMS VAX 8600 CONSOLE DISK DRIVER  
0000 2 .IDENT 'V04-001'  
0000 3 .  
0000 4 \*\*\*\*\*  
0000 5 .  
0000 6 . COPYRIGHT (c) 1978, 1980, 1982, 1984 BY  
0000 7 . DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.  
0000 8 . ALL RIGHTS RESERVED.  
0000 9 .  
0000 10 . THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  
0000 11 . ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE  
0000 12 . INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER  
0000 13 . COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  
0000 14 . OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY  
0000 15 . TRANSFERRED.  
0000 16 .  
0000 17 . THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  
0000 18 . AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT  
0000 19 . CORPORATION.  
0000 20 .  
0000 21 . DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  
0000 22 . SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.  
0000 23 .  
0000 24 .  
0000 25 \*\*\*\*\*  
0000 26 .  
0000 27 . FACILITY:  
0000 28 .  
0000 29 . VAX/VMS VAX 8600 CONSOLE RL02 DRIVER  
0000 30 .  
0000 31 . AUTHOR:  
0000 32 .  
0000 33 . BENN SCHREIBER, 15-MAR-1983  
0000 34 .  
0000 35 . MODIFIED BY:  
0000 36 .  
0000 37 . V04-001 BLS0345 Benn Schreiber 27-AUG-1984  
0000 38 . Retry complete transfer rather than attempting restart  
0000 39 . at last block. This avoids forking per-block in the  
0000 40 . non-error case. Wait for ready before issuing sts/reset  
0000 41 . on error path. Increase timeout on read/write operations.  
0000 42 . Check for errors on get status interrupts following read/write.  
0000 43 .  
0000 44 . V03-005 BLS0342 Benn Schreiber 19-AUG-1984  
0000 45 . Implement abort, reset with status. Modify cvc\_getsts  
0000 46 . to use TIMEDWAI macro.  
0000 47 .  
0000 48 . V03-004 TCM0002 Trudy C. Matthews 09-Aug=1984  
0000 49 . Increase timeout value in CVC\_GETSTS from 30 to 100000.  
0000 50 .  
0000 51 . V03-003 TCM0001 Trudy C. Matthews 08-Aug-1984  
0000 52 . Several bug fixes. Also a spec change - the LBN in STXCS  
0000 53 . must always be valid for each word of the transfer (and not  
0000 54 . just the first).  
0000 55 .  
0000 56 . . .

0000 58 : ABSTRACT:  
0000 59 :  
0000 60 : THIS MODULE CONTAINS THE TABLES AND ROUTINES NECESSARY TO  
0000 61 : PERFORM ALL DEVICE-DEPENDENT PROCESSING OF AN I/O REQUEST  
0000 62 : FOR RL02 DISK TYPES ON A VAX/VMS VAX 8600 CONSOLE SUBSYSTEM.  
0000 63 :  
0000 64 : THE DISKS HAVE THE FOLLOWING PHYSICAL GEOMETRY:  
0000 65 :  
0000 66 : # CYL TRACKS/ CYLINDER SECTORS/ TRACK BYTES/ SECTOR MAXIMUM  
0000 67 :  
0000 68 :  
0000 69 : RL02 512 2 40 256 20480  
0000 70 :  
0000 71 : THE IOSX INHSEEK FUNCTION MODIFIER IS TREATED AS A NO-OP BY  
0000 72 : THIS DRIVER, SINCE AN IMPLICIT SEEK IS ALWAYS DONE BY THE  
0000 73 : CONSOLE SUBSYSTEM WHEN READING/WRITING.  
0000 74 :  
0000 75 :--

```

0000 77 .SBTTL EXTERNAL AND LOCAL DEFINITIONS
0000 78
0000 79 :
0000 80 : EXTERNAL SYMBOLS
0000 81 :
0000 82
0000 83 SCRBDDEF :DEFINE CHANNEL REQUEST BLOCK
0000 84 SDCDEF :DEFINE DEVICE CLASS
0000 85 SDBBDEF :DEFINE DEVICE DATA BLOCK
0000 86 SDEVDEF :DEFINE DEVICE CHARACTERISTICS
0000 87 SDPTDEF :DEFINE DRIVER PROLOGUE TABLE
0000 88 SDYNDEF :DEFINE DYNAMIC DATA STRUCTURE TYPES
0000 89 SEMBDEF :DEFINE ERROR MESSAGE BUFFER
0000 90 SIDBDEF :DEFINE INTERRUPT DATA BLOCK
0000 91 SIODEF :DEFINE I/O FUNCTION CODES
0000 92 SIRPDEF :DEFINE I/O REQUEST PACKET
0000 93 SPRDEF :DEFINE PROCESSOR REGISTERS
0000 94 SSSDEF :DEFINE SYSTEM STATUS CODES
0000 95 SUCBDEF :DEFINE UNIT CONTROL BLOCK
0000 96 SVECDEF :DEFINE INTERRUPT VECTOR BLOCK
0000 97
0000 98 :
0000 99 : GENF
0000 100 : GENERATE CASE TABLE INDEX SYMBOL
0000 101 .MACRO GENF FCODE
0000 102 CD'FCODE=$$GENF_CODE
0000 103 $$GENF_CODE=$$GENF_CODE+1
0000 104 .ENDM
0000 105 :
0000 106 : LOCAL SYMBOLS
0000 107 :
0000 108 CV_NUM_REGS = 4 :NUMBER OF DEVICE REGISTERS (MIMIC RL02)
0000 109 CV_SLM = 5 :STATE=SEEK LINEAR MODE (READY TO GO)
0000 110 :
0000 111 : UCB OFFSETS WHICH FOLLOW THE STANDARD UCB FIELDS
0000 112 :
0000 113 SDEFINI UCB ;START OF UCB DEFINITIONS
0000 114
0000 115 .=UCBSK_LCL_DISK_LENGTH :BEGIN DEFINITIONS AT END OF UCB
00CC 116 $DEF UCBSL_CV-CS .BLKL 1 :CONTROL STATUS REGISTER
00D0 117 $DEF UCBSL_CV-MP .BLKL 1 :MULTIPURPOSE REGISTER
00D4 118 $DEF UCBSQ_CV-CSMP .BLKQ 1 :SAVE CS AND MP DURING RESET STATUS
00DC 119 $DEF UCBSB_CV-STATE .BLKB 1 :CURRENT INTERRUPT STATE **ADJACENCY
00DD 120 $DEF UCBSB_CV-STS .BLKB 1 :STATUS FLAGS **ASSUMED
00DE 121 $DEF UCBSW_CV-BBC .BLKW 1 :BLOCK BYTE COUNT REMAINING
00E0 122 $DEF UCBSL_CV-IBUF .BLKL 1 :INTERNAL BUFFER FOR READING
00E4 123 $DEF UCBSL_CV-MVRTN .BLKL 1 :ADDRESS OF BUFFER MOVE ROUTINE
00E8 124 $DEF UCBSL_CV-BUFWIN .BLKL 1 :BUFFER WINDOW
00EC 125 $DEF UCBSQ_CV-BDAT .BLKQ 1 :SAVPTA AND TRANSFER PARAMS THIS BLOCK
00F4 126 $DEF UCBSL_CV-LBN .BLKL 1 :SAVE STARTING LBN OF TRANSFER
00FB 127 $DEF UCBSL_CV-ABPC .BLKL 1 :SAVE RETURN ADDRESS FROM ABORT CALL
00FC 128 $DEF UCBSK_CV-LEN .BLKW 0 :LENGTH OF UCB
00FC 129 SDEFEND OCB :END OF UCB DEFINITONS
0000 130
0000 131 :
0000 132 : RL11/RL01 REGISTER OFFSETS FROM CSR ADDRESS
0000 133 :

```

```

0000 134 SDEFINI CV : START OF REGISTER DEFINITIONS
0000 135
0000 136
0000 137 ; UCBSSB_CV_STS FLAGS
0000 138 ; -VIELD CV,0,<-  

0000 140 <RD,,M>,- :SET IF READ OPERATION
0000 141 <STSONLY,,M>,- :OPERATION IS GET STATUS ONLY
0000 142 <STSError,,M>,- :ERROR FROM CONSOLE ON GETSTS INTERRUPT
0000 143 <ABORT,,M>> :ABORT CURRENT OPERATION AND RETRY
0000 144
0000 145 SDEF CV CS .BLKW 1 :CONTROL STATUS REGISTER (CSR)
0002 146 -VIELD CV_CS,0,<- :START OF CSR BIT DEFINITIONS
0002 147 <DRDY,,M>,- :DRIVE READY
0002 148 <,3>,- :FUNCTION CODE
0002 149 <,2>,- :BUS ADDRESS EXTENSION BITS
0002 150 <,1>,- :INTERRUPT ENABLE
0002 151 <,1>,- :CONTROLLER READY
0002 152 <DS,2>,- :DRIVE SELECT
0002 153 <OPI,,M>,- :OPERATION INCOMPLETE
0002 154 <CRC,,M>,- :DATA CRC OR HEADER CRC
0002 155 <CVT,,M>,- :DATA LATE OR HEADER NOT FOUND
0002 156 <NXM,,M>,- :NON-EXISTENT MEMORY
0002 157 <DE,,M>,- :DRIVE ERROR
0002 158 <CE,,M>,- :COMPOSITE ERROR
0002 159 > :END CSR BIT DEFINITIONS
0002 160
0002 161 SDEF CV MP .BLKW 1 :MULTIPURPOSE REGISTER (MPR)
0004 162 -VIELD CV_MP,0,<- :START OF MPR BIT DEFINITIONS
0004 163 <STA,3>,- :DRIVE STATE
0004 164 <BH,,M>,- :BRUSH HOME
0004 165 <HO,,M>,- :HEADS OUT
0004 166 <CO,,M>,- :COVER OPEN
0004 167 <HS,,M>,- :HEAD SELECT
0004 168 <,1>,- :DRIVE TYPE
0004 169 <DSE,,M>,- :DRIVE SELECT ERROR
0004 170 <VC,,M>,- :VOLUME CHECK
0004 171 <WGE,,M>,- :WRITE GATE ERROR
0004 172 <SPE,,M>,- :SPIN ERROR
0004 173 <SKTO,,M>,- :SEEK TIME OUT
0004 174 <WL,,M>,- :WRITE LOCK
0004 175 <CHE,,M>,- :CURRENT HEAD ERROR
0004 176 <WDE,,M>,- :WRITE DATA ERROR
0004 177 > :END MPR BIT DEFINITIONS
0004 178
0004 179 ; VAX 8600 STXCS FORMAT
0004 180 ; -VIELD STXCS,0,<-
0004 181 <FUNC,4>,- :DISK FUNCTION TO PERFORM
0004 182 <,2>,- :MBZ
0004 183 <IE,,M>,- :INTERRUPT ENABLE
0004 184 <RDY,,M>,- :READY
0004 185 <ADDR$16>,- :DISK LOGICAL BLOCK NUMBER
0004 186 <STS,8>,- :STATUS OF TRANSFER
0004 187
0004 188
0004 189
0004 190

```

0004	191	SDEFEND CV	:END RL11/RL01 REGISTER DEFINITIONS	
	0000	192		
	0000	193		
	0000	194	: VAX 8600 CONSOLE STXCS STATUS CODES	
	0000	195	:	
00000001	0000	196	TRANS_COMPLETE = 1	: TRANSACTION COMPLETED
00000002	0000	197	TRANS_CONTINUE = 2	: CONTINUE TRANSACTION
00000003	0000	198	TRANS_ABORTED = 3	: TRANSACTION ABORTED
00000004	0000	199	RETURNED_STATUS = 4	: STATUS RETURNED
00000080	0000	200	HANDSHAKE_ERROR = ^X80	: HANDSHAKE ERROR DURING TRANSACTION
00000081	0000	201	HW_ERROR = ^X81	: HARDWARE ERROR DURING TRANSACTION
	0000	202	:	
	0000	203	: VAX 8600 CONSOLE STXCS FUNCTION CODES	
	0000	204	:	
00000000	0000	205	NO_OP = 0	: NO OPERATION
00000002	0000	206	STATUS_RESET = 2	: READ DEVICE STATUS WITH RST ASSERTED
00000003	0000	207	ABORT_TRANSFER = 3	: ABORT CURRENT TRANSFER
00000004	0000	208	READ_STATUS = 4	: READ DEVICE STATUS
00000005	0000	209	WRITE_BLOCK = 5	: WRITE BLOCK OF DATA
00000006	0000	210	READ_BLOCK = 6	: READ BLOCK OF DATA
	0000	211	:	
	0000	212	: INTERRUPT TRANSITION CODES	
	0000	213	:	
00000000	0000	214	ITC_DATA = 0	: READ OR WRITE DATA
00000001	0000	215	ITC_STS1 = 1	: GET CONTROL/STATUS REGISTER
00000002	0000	216	ITC_STS2 = 2	: GET RL11 MULTIPURPOSE REGISTER
00000003	0000	217	ITC_ABORT = 3	: ABORT CURRENT TRANSFER
00000004	0000	218	ITC_RESET1 = 4	: GET CONTROL/STATUS WITH RST ASSERTED
00000005	0000	219	ITC_RESET2 = 5	: GET MP REG WITH RST ASSERTED
	0000	220	:	
	0000	221	: DEFINE THE CASE OFFSETS AS CD'FUNCTION (I.E. CDF_NOP)	
	0000	222	:	
00000000	0000	223	\$SGENF_CODE = 0	: INITIALIZE
	0000	224	GENF F_NOP	: NO-OP
	0000	225	GENF F_UNLOAD	: UNLOAD VOLUME (NOP)
	0000	226	GENF F_SEEK	: SEEK
	0000	227	GENF F_RECAL	: RECALIBRATE (NOP)
	0000	228	GENF F_DRVCLR	: DRIVE CLEAR (RESET & GET STATUS)
	0000	229	GENF F_RELEASE	: RELEASE PORT (NOP)
	0000	230	GENF F_OFFSET	: OFFSET HEADS (NOP)
	0000	231	GENF F_RETCENTER	: RETURN HEADS TO CENTERLINE (NOP)
	0000	232	GENF F_PACKACK	: PACK ACKNOWLEDGE (RESET & GET STATUS)
	0000	233	GENF F_SEARCH	: SEARCH (NOP)
	0000	234	GENF F_WRITECHECK	: WRITE CHECK
	0000	235	GENF F_WRIITEDATA	: WRITE DATA
	0000	236	GENF F_READDATA	: READ DATA
	0000	237	GENF F_WRITEHEAD	: WRITE HEADERS (NOP)
	0000	238	GENF F_READHEAD	: READ HEADERS
	0000	239	GENF F_NOP	: place holder
	0000	240	GENF F_NOP	: place holder
	0000	241	GENF F_AVAILABLE	: AVAILABLE
	0000	242		

```

0000 244 .SBTTL STANDARD TABLES
0000 245
0000 246
0000 247 : DRIVER PROLOGUE TABLE
0000 248 :
0000 249 : THE DPT DESCRIBES DRIVER PARAMETERS AND I/O DATABASE FIELDS
0000 250 : THAT ARE TO BE INITIALIZED DURING DRIVER LOADING AND RELOADING
0000 251 :
0000 252
0000 253 DPTAB - :DPT CREATION MACRO
0000 254 END=CV-END,- :END OF DRIVER LABEL
0000 255 ADAPTER=UBA,- :ADAPTER TYPE = UNIBUS
0000 256 FLAGS=DPTSM-SVP,- :SYSTEM PAGE TABLE ENTRY REQUIRED
0000 257 UCBSIZE=UCBSK-CV-LEN,- :LENGTH OF UCB
0000 258 NAME=CVDRIVER :DRIVER NAME
0038 259
0038 260 DPT_STORE INIT :START CONTROL BLOCK INIT VALUES
0038 261 DPT_STORE DDB, DDBSL-ACPD, L, <"A\FL11> :DEFAULT ACP NAME
003F 262 DPT_STORE DDB, DDBSL-ACPD+3, B, DDBSK CART :ACP CLASS
0043 263 DPT_STORE UCB, UCBSB-FIPL, B, 8 :FORK IPL
0047 264 DPT_STORE UCB, UCBSL-DEVCHAR, L, - :DEVICE CHARACTERISTICS
0047 265 <DEVSM-FOD- :FILES ORIENTED
0047 266 !DEVSM-DIR- :DIRECTORY STRUCTURED
0047 267 !DEVSM-AVL- :AVAILABLE
0047 268 !DEVSM-ELG- :ERROR LOGGING
0047 269 !DEVSM-SHR- :SHAREABLE
0047 270 !DEVSM-IDV- :INPUT DEVICE
0047 271 !DEVSM-ODV- :OUTPUT DEVICE
0047 272 !DEVSM-RND> :RANDOM ACCESS
004E 273 DPT_STORE UCB, UCBSB-DEVCLASS, B, DCS-DISK :DEVICE CLASS
0052 274 DPT_STORE UCB, UCBSW-DEBUFSIZ, W, 512 :DEFAULT BUFFER SIZE
0057 275 DPT_STORE UCB, UCBSB-SECTORS, B, 40 :NUMBER OF SECTORS PER TRACK
005B 276 DPT_STORE UCB, UCBSB-TRACKS, B, 2 :NUMBER OF TRACKS PER CYLINDER
005F 277 DPT_STORE UCB, UCBSB-DIPL, B, 21 :DEVICE IPL
0063 278 DPT_STORE UCB, UCBSB-ERTMAX, B, 8 :MAX ERROR RETRY COUNT
0067 279
0067 280 DPT_STORE REINIT :START CONTROL BLOCK RE-INIT VALUES
0067 281 DPT_STORE CRB, CRBSL-INTD+4, D, CV INT :INTERRUPT SERVICE ROUTINE ADDRESS
006C 282 DPT_STORE CRB, CRBSL-INTD+VECSL-INITIAL, - :CONTROLLER INIT ADDRESS
006C 283 D, CV RL11 INIT
0071 284 DPT_STORE CRB, CRBSL-INTD+VECSL-UNITINIT, - :UNIT INIT ADDRESS
0071 285 D, CV RLOX INIT
0076 286 DPT_STORE DDB, DDBSL-DDT, D, CVSDDT :DDT ADDRESS
007B 287
007B 288 DPT_STORE END :END OF INITIALIZATION TABLE
0000 289
0000 290
0000 291 : DRIVER DISPATCH TABLE
0000 292 :
0000 293 : THE DDT LISTS ENTRY POINTS FOR DRIVER SUBROUTINES WHICH ARE
0000 294 : CALLED BY THE OPERATING SYSTEM.
0000 295 :
0000 296
0000 297 DDTAB - :DDT CREATION MACRO
0000 298 DEVNAM=CV,- :NAME OF DEVICE
0000 299 START=CV-STARTIO,- :START I/O ROUTINE
0000 300 UNSOLIC=CV-UNSOLONT,- :UNSOLICITED INTERRUPT

```

0000 301 FUNCTB=CV\_FUNCTABLE,- :FUNCTION DECISION TABLE  
0000 302 CANCEL=0,- :CANCEL=NO-OP FOR FILES DEVICE  
0000 303 REGDMP=CV\_REGDUMP,- :REGISTER DUMP ROUTINE  
0000 304 DIAGBF=<<CV\_NUM\_REGS+5+5+3+1>>\*4>,- :BYTES IN DIAG BUFFER  
0000 305 ERLGBF=<<<CV\_NUM\_REGS+5+1>>\*4>+EMBSL\_DV\_REGS>> :BYTES IN  
0038 306 ;ERROR LOG BUFFER  
0038 307  
0038 308 : DIAGNOSTIC BUFFER SIZE = <<4 RL02 REGISTER LONGWORDS + 5 UCB FIELD LONGWORDS  
0038 309 + 5 IOC\$DIAGBUFLILL LONGWORDS + 3 BUFFER ALLOCATION  
0038 310 LONGWORDS + 1 LONGWORD FOR # REGISTERS IN CV\_REGDUMP>  
0038 311 \* 4 BYTES/LONGWORD>  
0038 312  
0038 313 : ERROR LOG BUFFER SIZE = <<<4 RL02 REGISTER LONGWORDS + 5 UCB FIELD LONGWORDS  
0038 314 + 1 LONGWORD FOR # REGISTERS IN CV\_REGDUMP>  
0038 315 \* 4 BYTES/LONGWORD> + BYTES NEEDED FOR ERROR LOGGER  
0038 316 TO SAVE SOFTWARE REGISTERS>  
0038 317  
0038 318

0038 320 : FUNCTION DECISION TABLE

0038 321 : THE FDT LISTS VALID FUNCTION CODES, SPECIFIES WHICH  
0038 322 : CODES ARE BUFFERED, AND DESIGNATES SUBROUTINES TO  
0038 323 : PERFORM PREPROCESSING FOR PARTICULAR FUNCTIONS.

0038 324 :  
0038 325 :  
0038 326 :  
0038 327 :  
0038 328 CV\_FUNCTABLE:  
0038 329 FUNCTAB , -  
0038 330 <NOP,- : LIST LEGAL FUNCTIONS  
0038 331 UNLOAD,- : NO-OP  
0038 332 SEEK,- : UNLOAD  
0038 333 DRVCLR,- : SEEK  
0038 334 PACKACK,- : DRIVE CLEAR  
0038 335 SENSECHAR,- : PACK ACKNOWLEDGE  
0038 336 SETCHAR,- : SENSE CHARACTERISTICS  
0038 337 SENSEMODE,- : SET CHARACTERISTICS  
0038 338 SETMODE,- : SENSE MODE  
0038 339 READLBLK,- : SET MODE  
0038 340 WRITELBLK,- : READ LOGICAL BLOCK  
0038 341 READPBLK,- : WRITE LOGICAL BLOCK  
0038 342 WRITEPBLK,- : READ PHYSICAL BLOCK  
0038 343 READVBLK,- : WRITE PHYSICAL BLOCK  
0038 344 WRITEVBLK,- : READ VIRTUAL BLOCK  
0038 345 AVAILABLE,- : WRITE VIRTUAL BLOCK  
0038 346 ACCESS,- : AVAILABLE  
0038 347 ACPCONTROL,- : ACCESS FILE / FIND DIRECTORY ENTRY  
0038 348 CREATE,- : ACP CONTROL FUNCTION  
0038 349 DEACCESS,- : CREATE FILE AND/OR DIRECTORY ENTRY  
0038 350 DELETE,- : DEACCESS FILE  
0038 351 MODIFY,- : DELETE FILE AND/OR DIRECTORY ENTRY  
0038 352 MOUNT- : MODIFY FILE ATTRIBUTES  
0038 353 >  
0040 354 FUNCTAB , - : MOUNT VOLUME  
0040 355 <NOP,- : BUFFERED FUNCTIONS  
0040 356 UNLOAD,- : NO-OP  
0040 357 SEEK,- : UNLOAD  
0040 358 DRVCLR,- : SEEK  
0040 359 PACKACK,- : DRIVE CLEAR  
0040 360 SENSECHAR,- : PACK ACKNOWLEDGE  
0040 361 SETCHAR,- : SENSE CHARACTERISTICS  
0040 362 SENSEMODE,- : SET CHARACTERISTICS  
0040 363 SETMODE,- : SENSE MODE  
0040 364 AVAILABLE,- : SET MODE  
0040 365 ACCESS,- : AVAILABLE  
0040 366 ACPCONTROL,- : ACCESS FILE / FIND DIRECTORY ENTRY  
0040 367 CREATE,- : ACP CONTROL FUNCTION  
0040 368 DEACCESS,- : CREATE FILE AND/OR DIRECTORY ENTRY  
0040 369 DELETE,- : DEACCESS FILE  
0040 370 MODIFY,- : DELETE FILE AND/OR DIRECTORY ENTRY  
0040 371 MOUNT- : MODIFY FILE ATTRIBUTES  
0040 372 >  
0048 373 FUNCTAB CV ALIGN,- : MOUNT VOLUME  
0048 374 <READLBLK,- : TEST ALIGNMENT FUNCTIONS  
0048 375 READPBLK,- : READ LOGICAL BLOCK  
0048 376 READVBLK,- : READ PHYSICAL BLOCK  
0048 > : READ VIRTUAL BLOCK

0048	377	WRITELBLK,-	: WRITE LOGICAL BLOCK
0048	378	WRITEPBLK,-	: WRITE PHYSICAL BLOCK
0048	379	WRITEVBLK-	: WRITE VIRTUAL BLOCK
0048	380	>	
0054	381	FUNCTAB +ACPSREADBLK,-	: READ FUNCTIONS
0054	382	<READLBLK,-	: READ LOGICAL BLOCK
0054	383	READPBLK,-	: READ PHYSICAL BLOCK
0054	384	READVBLK-	: READ VIRTUAL BLOCK
0054	385	>	
0060	386	FUNCTAB +ACPSWRITEBLK,-	: WRITE FUNCTIONS
0060	387	<WRITELBLK,-	: WRITE LOGICAL BLOCK
0060	388	WRITEPBLK,-	: WRITE PHYSICAL BLOCK
0060	389	WRITEVBLK-	: WRITE VIRTUAL BLOCK
0060	390	>	
006C	391	FUNCTAB +ACPSACCESS,-	: ACCESS FUNCTIONS
006C	392	<ACCESS,-	: ACCESS FILE / FIND DIRECTORY ENTRY
006C	393	CREATE-	: CREATE FILE AND/OR DIRECTORY ENTRY
006C	394	>	
0078	395	FUNCTAB +ACPSDEACCESS,-	: DEACCESS FUNCTION
0078	396	<DEACCESS-	: DEACCESS FILE
0078	397	>	
0084	398	FUNCTAB +ACPSMODIFY,-	: MODIFY FUNCTIONS
0084	399	<ACPCONTROL,-	: ACP CONTROL FUNCTION
0084	400	DELETE,-	: DELETE FILE AND/OR DIRECTORY ENTRY
0084	401	MODIFY-	: MODIFY FILE ATTRIBUTES
0084	402	>	
0090	403	FUNCTAB +ACPSMOUNT,-	: MOUNT FUNCTION
0090	404	<MOUNT-	: MOUNT VOLUME
0090	405	>	
009C	406	FUNCTAB +EXESLCLDISKVALID,-	
009C	407	<UNLOAD,-	
009C	408	AVAILABLE,-	
009C	409	PACKACK-	
009C	410	>	
00A8	411	FUNCTAB +EXESZEROPARM,-	: ZERO PARAMETER FUNCTIONS
00A8	412	<NOP,-	: NO-OP
00A8	413	UNLOAD,-	: UNLOAD
00A8	414	DRVCLR,-	: DRIVE CLEAR
00A8	415	PACKACK,-	: PACK ACKNOWLEDGE
00A8	416	AVAILABLE,-	: AVAILABLE
00A8	417	>	
00B4	418	FUNCTAB +EXESONEPARM,-	: ONE PARAMETER FUNCTION
00B4	419	<SEEK-	: SEEK
00B4	420	>	
00C0	421	FUNCTAB +EXESSENSEMODE,-	: SENSE FUNCTIONS
00C0	422	<SENSECHAR,-	: SENSE CHARACTERISTICS
00C0	423	SENSEMODE-	: SENSE MODE
00C0	424	>	
00CC	425	FUNCTAB +EXESSETCHAR,-	: SET FUNCTIONS
00CC	426	<SETCHAR,-	: SET CHARACTERISTICS
00CC	427	SETMODE-	: SET MODE
00CC	428	>	

00D8 470 .SBTTL CONTROLLER INITIALIZATION ROUTINE  
00D8 471 :  
00D8 472 :++  
00D8 473 :  
00D8 474 : FUNCTIONAL DESCRIPTION:  
00D8 475 :  
00D8 476 : THIS ROUTINE IS A NO-OP FOR THE RL11 BUT MUST BE INCLUDED  
00D8 477 : SINCE IT IS CALLED WHEN THE RL02 IS BOOTTED AS A SYSTEM DEVICE.  
00D8 478 :  
00D8 479 : THE OPERATING SYSTEM CALLS THIS ROUTINE:  
00D8 480 : - AT SYSTEM STARTUP  
00D8 481 : - DURING DRIVER LOADING  
00D8 482 : - DURING RECOVERY FROM POWER FAILURE  
00D8 483 :  
00D8 484 : INPUTS:  
00D8 485 :  
00D8 486 : R4 - CSR ADDRESS (DEVICE CONTROL STATUS REGISTER)  
00D8 487 : RS - IDB ADDRESS (INTERRUPT DATA BLOCK)  
00D8 488 : ALL INTERRUPTS ARE LOCKED OUT  
00D8 489 :  
00D8 490 : OUTPUTS:  
00D8 491 :  
00D8 492 : CONTROL IS RETURNED TO THE CALLER.  
00D8 493 :  
00D8 494 :--  
00D8 495 :  
05 00D8 496 CV\_RL11\_INIT:  
00D8 497 RSB :CONTROLLER INITIALIZATION  
:RETURN TO CALLER

00D9 459 .SBTTL UNIT INITIALIZATION ROUTINE  
 00D9 460  
 00D9 461 :++  
 00D9 462 : CV\_RLOX\_INIT - UNIT INITIALIZATION ROUTINE  
 00D9 463 :  
 00D9 464 : FUNCTIONAL DESCRIPTION:  
 00D9 465 :  
 00D9 466 : THIS ROUTINE READIES THE RL02 UNIT FOR I/O OPERATIONS.  
 00D9 467 :  
 00D9 468 :  
 00D9 469 : THE OPERATING SYSTEM CALLS THIS ROUTINE:  
 00D9 470 : - AT SYSTEM STARTUP  
 00D9 471 : - DURING DRIVER LOADING  
 00D9 472 : - DURING RECOVERY FROM POWER FAILURE  
 00D9 473 :  
 00D9 474 : INPUTS:  
 00D9 475 :  
 00D9 476 : R4 - CSR ADDRESS (CONTROLLER STATUS REGISTER)  
 00D9 477 : RS - UCB ADDRESS (UNIT CONTROL BLOCK)  
 00D9 478 :  
 00D9 479 : OUTPUTS:  
 00D9 480 :  
 00D9 481 : THE DRIVE UNIT IS RESET, UCB FIELDS ARE INITIALIZED, AND THE  
 00D9 482 : ROUTINE WAITS FOR ONLINE UNITS TO SPIN UP. ALL REGISTERS  
 00D9 483 : EXCEPT R0-R3 ARE PRESERVED.  
 00D9 484 :  
 00D9 485 :--  
 00D9 486 :  
 00D9 487 CV\_RLOX\_INIT: :RL01/RL02 UNIT INITIALIZATION  
 0810 8F AA 00D9 488 BICW2 #UCBSM\_ONLINE!UCBSM\_VALID,- ;ASSUME OFFLINE/INVALID  
 64 A5 00D9 489 UCBSW\_STS(R5) ;...  
 2324C002 8F DD 00DF 490 MOVL #^X2324C002,- ;SET MEDIA IDENT 'DL RL02'  
 008C C5 00E5 491 UCBSL\_MEDIA\_ID(R5)  
 0A 90 00E8 492 MOVB S#DT\$\_RL02,- ;SET RL02 DEVICE TYPE  
 41 A5 00EA 493 UCB\$B\_DEVTYPE(R5)  
 46 A5 0200 8F B0 00EC 494 MOVW #512\_UCBSW\_CYINDERS(R5);SET NUMBER OF RL02 CYLINDERS  
 00B0 C5 5000 8F 3C 00F2 495 MOVZWL #20480\_UCBSL\_MAXBLOCK(R5);SET MAX RL02 BLOCK NUMBER  
 OF 10 00F9 496 BSBB CVC\_GESTS ;GET CONSOLE RL02 STATUS  
 06 50 00 E1 00FB 497 BBC #CV\_CS\_V\_DRDY,R0,403 ;BRANCH IF DRIVE NOT READY  
 64 A5 0800 8F A8 00FF 498 BISW2 #UCBSM\_VALID,UCBSW\_STS(R5) ;YES, SET VOLUME VALID  
 64 A5 10 A8 0105 499 BISW2 #UCBSM\_VALID,UCBSW\_STS(R5) ;SET UNIT ONLINE  
 05 0109 500 608: RSB

010A 503 .SBTTL DRIVER SPECIFIC SUBROUTINES  
 010A 504 .  
 010A 505 : CVC\_GETSTS - GET STATUS FOR VAX 8600 CONSOLE RL02 WITHOUT INTERRUPTS  
 010A 506 :  
 010A 507 : INPUTS:  
 010A 508 :  
 010A 509 : NONE  
 010A 510 :  
 010A 511 : OUTPUTS:  
 010A 512 :  
 010A 513 : R0 = 0 IF FAILED TO GET STATUS  
 010A 514 : = RL02 CONTROL STATUS REGISTER  
 010A 515 :  
 010A 516 : R1 = RL02 MULTIPURPOSE REGISTER (UNUSABLE IF R0=0)  
 010A 517 :  
 010A 518 : CVC\_GETSTS:  
 S2 DD 010A 519 PUSHL R2 ;SAVE R2  
 0D 10 010C 520 BSBB 100\$ ;READ CONTROL STATUS REGISTER  
 50 DD 010E 521 PUSHL R0 ;SAVE R0  
 09 10 0110 522 BSBB 100\$ ;READ MULTIPURPOSE REGISTER  
 S1 50 00 0112 523 MOVL R0,R1 ;POSITION MULTIPURPOSE REGISTER  
 50 BE DD 0115 524 POPL R0 ;RESTORE CSR  
 04 BA 0118 525 POPR #^M<R2> ;RESTORE R2  
 05 011A 526 RSB ;  
 0000004C 8F 02 DA 011B 527 ;  
 011B 528 100\$: MTPR #STATUS RESET #PRS\_STXCS ;REQUEST READ STATUS  
 0122 529 TIMEDWAIT TIME=#600\*1000,-  
 0122 530 INS1=<MFPR #PRS\_STXCS,R2> - ;READ STATUS  
 0122 531 INS2=<BBS #STXCS\_V RDY,R2,140\$> ;BRANCH IF READY  
 05 014F 532 RSB ;CONSOLE NEVER GOT READY (TIMEDWAIT  
 0150 533 ;CLEAR R0 ON FALL-OUT)  
 50 0000004D SE 04 CO 0150 534 140\$: ADDL2 #4,SP ;CLEAR TIMEDWAIT'S COUNTER FROM STACK  
 0F DB 0153 535 MFPR #PRS\_STXDB,R0 ;OBTAIN STATUS FROM CONSOLE  
 05 015A 536 RSB ;

CV  
VOA

```
0158 538 .SBTTL FDT ROUTINE - TEST TRANSFER BYTE COUNT ALIGNMENT
0158 539
0158 540 ++
0158 541
0158 542 DL_ALIGN - FDT ROUTINE TO TEST XFER BYTE COUNT
0158 543
0158 544 FUNCTIONAL DESCRIPTION:
0158 545
0158 546 THIS ROUTINE IS CALLED FROM THE FUNCTION DECISION TABLE DISPATCHER
0158 547 TO CHECK THE BYTE COUNT PARAMETER SPECIFIED BY THE USER PROCESS
0158 548 FOR AN EVEN NUMBER OF BYTES (WORD BOUNDARY).
0158 549
0158 550 INPUTS:
0158 551
0158 552 R3 - IRP ADDRESS (I/O REQUEST PACKET)
0158 553 R4 - PCB ADDRESS (PROCESS CONTROL BLOCK)
0158 554 R5 - UCB ADDRESS (UNIT CONTROL BLOCK)
0158 555 R6 - CCB ADDRESS (CHANNEL CONTROL BLOCK)
0158 556 R7 - BIT NUMBER OF THE I/O FUNCTION CODE
0158 557 R8 - ADDRESS OF FDT TABLE ENTRY FOR THIS ROUTINE
0158 558 4(AP) - ADDRESS OF FIRST FUNCTION DEPENDENT QIO PARAMETER
0158 559
0158 560 OUTPUTS:
0158 561
0158 562 IF THE QIO BYTE COUNT PARAMETER IS ODD, THE I/O OPERATION IS
0158 563 TERMINATED WITH AN ERROR. IF IT IS EVEN, CONTROL IS RETURNED
0158 564 TO THE FDT DISPATCHER.
0158 565
0158 566 --
0158 567
0158 568 CV_ALIGN:
0158 569 BLBS 4(AP),108 ;CHECK BYTE COUNT AT P1(AP)
0158 570 RSB ;IF LBS - ODD BYTE COUNT
015F 571 10S: MOVZWL #SSS_IBUFLEN,RO ;EVEN - RETURN TO CALLER
0160 572 JMP G^EXESABORTIO ;SET BUFFER ALIGNMENT STATUS
0165 573 ;ABORT I/O
```

0168 574 .SBTTL START I/O ROUTINE  
 0168 575  
 0168 576 :\*\*  
 0168 577 : CV\_STARTIO - START I/O ROUTINE  
 0168 578 :  
 0168 579 : FUNCTIONAL DESCRIPTION:  
 0168 580 :  
 0168 581 : THIS FORK PROCESS IS ENTERED FROM THE EXECUTIVE AFTER AN I/O REQUEST  
 0168 582 : PACKET HAS BEEN DEQUEUED, AND PERFORMS THE FOLLOWING:  
 0168 583 :  
 0168 584 :  
 0168 585 : - ACTIVATES THE CONSOLE AFTER SETTING UCB FIELDS, AND OBTAINING  
 0168 586 : CONTROLLER RESOURCES  
 0168 587 :  
 0168 588 : - WAITS FOR AN INTERRUPT  
 0168 589 :  
 0168 590 : - REGAINS CONTROL AFTER THE ISR SERVICES THE INTERRUPT, AND  
 0168 591 : - RE-ACTIVATES THE CONSOLE IF THE ORIGINAL FUNCTION  
 0168 592 : IS A RETRIABLE ERROR, OR  
 0168 593 : - COMPLETES THE I/O REQUEST BY RELEASING RESOURCES,  
 0168 594 : SETTING STATUS CODES, AND RETURNING TO THE EXECUTIVE.  
 0168 595 :  
 0168 596 : INPUTS:  
 0168 597 :  
 0168 598 R3 - IRP ADDRESS (I/O REQUEST PACKET)  
 0168 599 R5 - UCB ADDRESS (UNIT CONTROL BLOCK)  
 0168 600 :  
 0168 601 : OUTPUTS:  
 0168 602 :  
 0168 603 R0 - FIRST I/O STATUS LONGWORD: STATUS CODE & BYTES XFERED  
 0168 604 R1 - SECOND I/O STATUS LONGWORD: 0 FOR DISKS  
 0168 605 :  
 0168 606 : THE I/O FUNCTION IS EXECUTED.  
 0168 607 :  
 0168 608 : ALL REGISTERS EXCEPT R0-R4 ARE PRESERVED.  
 0168 609 :  
 0168 610 :--  
 0168 611 :  
 0168 612 CV\_STARTIO: :START I/O OPERATION  
 0168 613 :  
 0168 614 : PREPROCESS UCB FIELDS  
 0168 615 :  
 0168 616 : PREPROCESS:  
 0168 617 :  
 0168 618 : Convert the physical media address in IRPSL\_MEDIA to an LBN.  
 0168 619 : This is necessary because the console RL02 controller expects an LBN,  
 0168 620 : not a physical media address. The LBN is given by the formula:  
 0168 621 :  
 0168 622 : LBN = (CYLINDER\*(TRACKS/CYLINDER)+TRACK)\*(SECTORS/TRACK))+SECTOR  
 0168 623 :  
 S3 38 A3 DD 0168 624 MOVL IRPSL\_MEDIA(R3),- :Copy media address to UCB  
 008C C5 00 0168 625 UCBSL\_MEDIA(R5) :  
 S3 008C C5 DE 0171 626 MOVAL UCBSL\_MEDIA(R5),R3 :Get address of media address  
 50 83 9A 0176 627 MOVZBL (R3)+,R0 :Get SECTOR  
 51 83 9A 0179 628 MOVZBL (R3)+,R1 :Get TRACK  
 7E 83 3C 017C 629 MOVZWL (R3)+,-(SP) :Get CYLINDER  
 S3 65 A5 9A 017F 630 MOVZBL UCBSB\_TRACKS(R5),R3 :Get TRACKS/CYLINDER

					MULL2	(SP)+,R3	R3 = C*(T/C)
					ADDL2	R3 R1	R1 = C*(T/C)+T
					MOVZBL	UCBSB_SECTORS(R5),R3	Get SECTORS/TRACK
					MULL2	R3,R1	R1 = (C*(T/C)+T)*(S/T)
008C C5	53 44 A5	51 51	C4 0183	631	ADDL3	R1, R0, UCBSL MEDIA(R5)	CALCULATE AND STORE LBN
00F4 C5	50 51	C4 0180	632	MOVL	UCBSL MEDIA(R5), UCBSL_CV	LBN(R5) :SAVE STARTING LBN FOR RETRIES	
	008C C5	00	0196	633	MOVE	UCBSB_ERTMAX(R5),-	INITIALIZE ERROR RETRY COUNT
	0081 C5	90	0190	634		UCBSB_ERTCNT(R5)	
	0080 C5		01A1	635	MOVL	UCBSL_IRP(R5), R3	:GET IRP ADDRESS
009A C5	53 58 A5	20 A5	00	636	MOVW	IRPSW_FUNC(R3), UCBSW_FUNC(R5)	:SAVE FUNCTION CODE
	0092 C5	EF 00	01A8	637	EXTZV	#IRPSW_FCODE,-	:EXTRACT I/O FUNCTION CODE
	78 A5	90	01B0	638	MOVB	#IRPSW_FCODE, IRPSW_FUNC(R3), R1	
	00EC C5	7D	01B4	639	MOVQ	R1, UCBSB_FEX(R5)	:STORE FUNCTION DISPATCH INDEX
	02 A5	AA 02	01B8	640	BICW2	UCBSQ_CV_BDAT(R5),-	:SAVE TRANSFER PARAMETERS
	07	E1 02	01C1	641	BBC	#UCBSM_DIAGBUF,-	
04 2A A3	01C3	07	01C5	642		UCBSW_DEVSTS(R5)	:CLR DIAGNOSTIC BUFFER PRESENT
68 A5	AB 02	01C8	643			#IRPSW_DIAGBUF,-	:IF CLR - NO DIAG BUFFER
			01CC	644	BISW2	IRPSW_STS(R3), FDISPATCH	
			01CC	645		#UCBSM_DIAGBUF, UCBSW_DEVSTS(R5)	:SET DIAG BUFFER PRESENT
			01CC	646			
			01CC	647			
			01CC	648			
			01CC	649			
			01CC	650			
			01CC	651			
			01CC	652			
			01CC	653			
			01CC	654			
			01CC	655			
			01CC	656			
			01CC	657			
			01CC	658			
00C0 C5	7E A5	AE	01CC	659	MNEGW	UCBSW_BCNT(R5), UCBSU_BCR(R5)	:INIT NEG BYTES LEFT TO XFER
00EC C5	7D	01D2	660	MOVQ	UCBSQ_CV_BDAT(R5),-	:RESTORE TRANSFER PARAMETERS	
	78 A5	01D6	661		UCBSL_SVAPTE(R5)		
008C C5	00F4 C5	DO	01D8	662	MOVL	UCBSL_CV_LBN(R5), UCBSEL_MEDIA(R5)	:RESTORE STARTING LBN
			01DF	663			
			01DF	664	MOVL	UCBSL_IRP(R5), R3	:GET IRP ADDRESS
			08 E0	665	BBS	#IRPSW_PHYSIO,-	:IF SET - PHYSICAL I/O FUNCTION
	0D 2A A3	01E3	666			IRPSW_STS(R3), 10\$	
	08	E0 01E8	667		BBS	#UCBSW_VALID,-	:IF SET - VOLUME SOFTWARE VALID
	08 64 A5	01EA	668			UCBSW_STS(R5), 10\$	
50 0254 8F	3C 01ED	669			MOVZWL	#SSS_VOLINV,R0	:SET VOLUME INVALID STATUS
	0240	31 01F2	670		BRW	RESETXFR	:RESET BYTE COUNT AND EXIT
53 0092 C5	9A 01F5	671	10\$:		MOVZBL	UCBSB_FEX(R5), R3	:GET FUNCTION DISPATCH INDEX
00DC C5	B4 01FA	672			CLRW	UCBSB_CV_STATE(R5)	:CLEAR INTERRUPT STATE AND STATUS
00EB C5	D4 01FE	673			CLRL	UCBSL_CV_BUFWIN(R5)	:CLEAR BUFFER WINDOW
			0202	674	CASE	R3,<-	:DISPATCH TO FUNCTION HANDLING ROUTINE
			0202	675		NOP,-	:NOP
			0202	676		UNLOAD,-	:UNLOAD
			0202	677		SEEK,-	:SEEK
			0202	678		NOP,-	:RECALIBRATE (unsupported)
			0202	679		DRVCLR,-	:DRVCLR
			0202	680		NOP,-	:RELEASE PORT (unsupported)
			0202	681		NOP,-	:OFFSET HEADS (unsupported)
			0202	682		NOP,-	:RETURN TO CENTER (unsupported)
			0202	683		PACKACK,-	:PACK ACKNOWLEDGE
			0202	684		NOP,-	:SEARCH (unsupported)
			0202	685		WRITECHECK,-	:WRITE CHECK (unsupported)
			0202	686		WRITEDATA,-	:WRITE DATA
			0202	687		READDATA,-	:READ DATA

			0202	688	NOP,-	: WRITE HEADER (unsupported)	
			0202	689	NOP,-	: READ HEADER	
			0202	690	NOP,-	: place holder	
			0202	691	NOP,-	: place holder	
			0202	692	AVAILABLE-	: AVAILABLE	
			022A	693	>		
			022A	694	NOP:	: NO-OP	
			022A	695	WRITECHECK:	: WRITE CHECK	
			022A	696	SEEK:	: SEEK	
			022A	697	DRVCLR:	: DRIVE CLEAR (GET STATUS & RESET)	
		0092 CS	94	022A	698 DO_FUNCTION:	: SET FUNCTION	
	0097	30	022E	700	CLRB	UCBSB_FEX(R5)	
		20	0231	701	BSBW	FEXL	
		19	0232	702	.BYTE	RETRYERR--1	
			0234	703	BRB	NORMAL	
			0234	704		: DONE	
	64 AS	0800 BF	A8	0234	705	PACKACK:	: PACK ACKNOWLEDGE (GET STATUS & RESET)
			023A	706	BISW2	#UCBSM_VALID -	
		EE	11	023A	707		UCBSW_STS(R5)
			023C	708	BRB	DO_FUNCTION	
			023C	709		: Then go do hardware function.	
			023C	710	UNLOAD:	: UNLOAD	
			023C	711	AVAILABLE:	: AVAILABLE	
	64 AS	0800 BF	AA	023C	712	BICW2	#UCBSM_VALID -
			0242	713		UCBSW_STS(R5)	
		D9	11	0242	714	BRB	NORMAC
			0244	715		: and go complete operation without	
	0000 CS	01	88	0244	716	READDATA:	: any hardware interaction.
			0249	717	BISB	#CV_M_RD,UCBSB_CV_STS(R5)	
	007C	30	0249	718	WRITEDATA:	: SET READ FLAG	
		05	024C	719	BSBW	FEXL	
			024D	720	.BYTE	RETRYERR--1	
			024D	721	:	: EXECUTE FUNCTION	
			024D	722	OPERATON COMPLETION	: ERROR OFFSET	
			024D	723	:		
			024D	724			
	50	01	3C	024D	725	NORMAL:	: SUCCESSFUL OPERATION COMPLETE
		43	11	0250	726	MOVZWL	#SSS_NORMAL,RO
			0252	727	BRB	FUNCXT	
			0252	728		: SET NORMAL COMPLETION STATUS	
			0252	729	RETRYERR:	: FUNCTION EXIT	
	0080 CS	97	0252	730	DEC8	UCBSB_ERTCNT(R5)	
		03	13	0256	731	BEQL	FATALERR
	FF71	31	0258	732	BRW	FDISPATCH	
			0258	733		: RETRIABLE ERROR	
	50	0254 BF	3C	0258	734	FATALERR:	: ANY RETRIES LEFT?
		00CC CS	7D	0260	735	MOVZWL	#SSS_VOLINV,RO
	2C 52	09	E0	0265	736	MOVA	UCBSL_CV_CS(R5),R1
			0269	737	BBS	#CV_MP_V_VC,R2,FUNCXT	
	50	025C BF	3C	0269	738		: GET CS IN R1 AND MP IN R2
		04	0D	026E	739	MOVZWL	#SSS_WRLCK,RO
	1F 52	0A	E0	0272	740	BBC	#CV_MP_V_WL,R2,108
			0276	741	BBS	#CV_MP_V_WGE,R2,FUNCXT	
	50	01F4 BF	3C	0276	742		: IF CLR - VOLUME NOT WRITE LOCKED
			0276	743		: IF SET - WRITE GATE ERROR	
			0276	744	108:	: IF WL & WGE SET - WRITE LOCK ERROR	
			0276	745	MOVZWL	#SSS_PARITY,RO	
						: ASSUME PARITY ERROR STATUS	

16 51 0B EO 027B	745	BBS	#CV_CS_V_CRC,R1,FUNCXT	:IF SET - CRC ERROR
04 51 0A E1 027F	746			:OR DATAPATH PURGE ERROR
0E 51 0C EO 0283	747	BBC	#CV_CS_V_OPI,R1,208	:HEADER NOT FOUND ERROR?
	748	BBS	#CV_CS_V_CVT,R1,FUNCXT	:IF OPI AND CVT SET - YES
50 008C BF 3C 0287	750	208:	MOVZWL #SSS_DRVERR,RO	:ASSUME DRIVE ERROR STATUS
05 51 0E EO 029C	751	BBS	#CV_CS_V_DE,R1,FUNCXT	:IF SET - DRIVE ERROR
50 0054 BF 3C 0290	752	MOVZWL #SSS_CTRLERR,RO	:ASSUME CONTROLLER ERROR STATUS	
	753			:FUNCTION EXIT
	754			:SAVE FINAL REQUEST STATUS
50 DD 0295	755	PUSHL R0		:FILL DIAGNOSTIC BUFFER IF PRESENT
00000000 GF 16 0297	756	JSB G^IOCSDIAGBUFILL		:DRIVE RELATED FUNCTION?
0092 CS 0A 91 029D	757	CMPB #CDF_WRITECHECK,UCBSB_FEX(R5)		:IF GTRU - YES
	758	BGTRU 108		:DRIVE RELATED FUNCTION?
0092 CS 11 91 02A2	759	CMPB #CDF_AVAILABLE,UCBSB_FEX(R5)		:IF EQL - YES
	760	BEQL 108		:RETRIEVE ADDRESS OF IRP
53 58 AS DD 02AB	761	MOVL UCBSL_IRP(R5),R3		:CALCULATE BYTES TRANSFERRED
00C0 CS A1 02AF	762	ADDW3 UCBSW_BCR(R5) -		:RELEASE CHANNEL IF OWNED
02 AE 32 A3 02B3	763	IRPSW_BCNT(R3),2(SP)		
	764	RELCHAN		
	765	108:		
51 D4 02BD	766	CLRL R1		:CLEAR SECOND STATUS LONGWORD
50 BED0 02BF	767	POPL R0		:RETRIEVE FINAL REQUEST STATUS
	768	REQCOM		:COMPLETE REQUEST
	769			

02C8 771  
 02C8 772  
 02C8 773 FEXL - RL11 HARDWARE FUNCTION EXECUTION  
 02C8 774  
 02C8 775 THIS ROUTINE IS CALLED VIA A BSB WITH A BYTE IMMEDIATELY FOLLOWING THAT  
 02C8 776 SPECIFIES THE ADDRESS OF AN ERROR ROUTINE. ALL DATA IS ASSUMED TO HAVE BEEN  
 02C8 777 SET UP IN THE UCB BEFORE THE CALL. THE APPROPRIATE PARAMETERS ARE LOADED  
 02C8 778 INTO THE CONSOLE STXCS AND THE FUNCTION IS INITIATED. THE RETURN ADDRESS  
 02C8 779 IS STORED IN THE UCB AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE  
 02C8 780 INTERRUPT OCCURS, CONTROL IS RETURNED TO THE CALLER.  
 02C8 781  
 02C8 782  
 02C8 783 INPUTS:  
 02C8 784 R5 = DEVICE UNIT UCB ADDRESS  
 02C8 785  
 02C8 786 00(SP) = RETURN ADDRESS OF CALLER  
 02C8 787 04(SP) = RETURN ADDRESS OF CALLER'S CALLER  
 02C8 788  
 02C8 789 IMMEDIATELY FOLLOWING INLINE AT THE CALL SITE IS A BYTE WHICH CONTAINS  
 02C8 790 A BRANCH DESTINATION TO AN ERROR RETRY ROUTINE.  
 02C8 791  
 02C8 792 OUTPUTS:  
 02C8 793  
 02C8 794 THERE ARE FOUR EXITS FROM THIS ROUTINE:  
 02C8 795  
 02C8 796 1. SPECIAL CONDITION - THIS EXIT IS TAKEN IF A POWER FAILURE OCCURS  
 02C8 797 OR THE OPERATION TIMES OUT. IT IS A JUMP TO THE APPROPRIATE  
 02C8 798 ERROR ROUTINE. IN THE CASE OF A POWER FAILURE, THE RETRY  
 02C8 799 COUNT IS RESET AND RETRIES INITIATED. IN THE CASE OF A  
 02C8 800 TIMEOUT, THE RETRY COUNT IS DECREMENTED AND RETRIES INITIATED  
 02C8 801 IF RETRIES REMAIN.  
 02C8 802  
 02C8 803 2. FATAL ERROR - THIS EXIT IS TAKEN IF A FATAL CONTROLLER OR DRIVE  
 02C8 804 ERROR OCCURS OR IF ANY ERROR OCCURS AND ERROR RETRY IS EITHER  
 02C8 805 INHIBITED OR EXHAUSTED. IT IS A JUMP TO THE FATAL ERROR EXIT  
 02C8 806 ROUTINE.  
 02C8 807  
 02C8 808 3. RETRIABLE ERROR - THIS EXIT IS TAKEN IF A RETRIABLE CONTROLLER  
 02C8 809 OR DRIVE ERROR OCCURS AND ERROR RETRY IS NEITHER INHIBITED  
 02C8 810 NOR EXHAUSTED. IT CONSISTS OF TAKING THE ERROR BRANCH EXIT  
 02C8 811 SPECIFIED AT THE CALL SITE. RETRIES ARE ACCOMPLISHED BY  
 02C8 812 RESTARTING THE ENTIRE I/O OPERATION, RATHER THAN AT THE  
 02C8 813 BLOCK IN ERROR.  
 02C8 814  
 02C8 815 4. SUCCESSFUL OPERATION - THIS EXIT IS TAKEN IF NO ERRORS OCCUR  
 02C8 816 DURING THE OPERATION. IT CONSISTS OF A RETURN INLINE.  
 02C8 817  
 02C8 818 IN ALL CASES IF AN ERROR OCCURS, AN ATTEMPT IS MADE TO LOG THE ERROR.  
 02C8 819  
 02C8 820 IN ALL CASES FINAL DEVICE REGISTERS ARE RETURNED VIA THE UCB.  
 02C8 821  
 02C8 822  
 02C8 823  
 02C8 824 FEXL:  
 02C8 825  
 02C8 826  
 02C8 827

POPL UCBSL\_DPC(R5) :FUNCTION EXECUTOR  
 MOVL UCBSL\_CRB(R5),R0 :SAVE DRIVER PC VALUE  
 MOVL CRBSL\_INTD+VE\SL\_IDB(R0),R1 :GET ADDRESS OF PRIMARY CRB  
 :GET ADDRESS OF IDB

50 009C C5 BEND  
 51 24 AS DO  
 52 2C A0 DO

04 A1 55 D1 02D5 828 CMPL R5 IDBSL\_OWNER(R1) ;DOES THIS PROCESS OWN CHANNEL?  
   05 12 02D9 829 BNEQ 105 ;IF NEQ - NO  
   54 61 D0 02DB 830 MOVL IDBSL\_CSR(R1),R4 ;SET ASSIGNED CHANNEL CSR ADDRESS  
   06 11 02DE 831 BRB 208 ;REQUEST CHANNEL (RETURNS R4 = (CSR ADR))  
   09 53 91 02E6 832 10\$: REQPCHAN  
   39 1A 02E9 833 20\$: CMPB R3 #CDF\_SEARCH ;TRANSFER FUNCTION?  
   02EB 834 BGTRU XFER ;BRANCH IF YES  
   02EB 835 ;IMMEDIATE FUNCTION EXECUTION  
   02EB 836 ;FUNCTIONS INCLUDE:  
   02EB 837  
   02EB 838  
   02EB 839  
   02EB 840  
   02EB 841  
   02EB 842  
   02EB 843  
   02EB 844  
   02EB 845  
   02EB 846 INPUTS: RS - UCB ADDRESS  
   02EB 847  
   02EB 848  
   02EB 849  
   02EB 850  
   02EB 851  
   02EB 852  
   02EB 853  
   02EB 854 IMMED: ;IMMEDIATE FUNCTION EXECUTION  
   02EB 855 DSBINT ;DISABLE INTERRUPTS  
   06 64 A5 05 E1 02F1 856 BBC #UCBSV\_POWER,UCBSW\_STS(R5),20\$ ;BRANCH IF NOT POWERFAIL  
   008D 31 02F6 857 ENBINT ;POWER FAIL  
   02F9 858 BRW RETREG ;PROCESS POWER FAILURE  
   02FC 859 ;  
   00DD C5 02 88 02FC 860 20\$: BISB2 #CV\_M\_STSONLY\_UCBSB CV\_STS(R5) ;REQUEST STATUS ONLY  
   00DC C5 01 90 0301 861 MOVBL #ITC\_STS1,UCBSB CV STATE(R5) ;SET STATE TO GETSTS1  
   50 44 8F 9A 0306 862 MOVZBL #<READ STATUS!STXCS\_M\_IE>,R0 ;LOAD THE FUNCTION  
   0000004C BF 50 DA 030A 863 MTPR R0,#PRS STXCS ;REQUEST STATUS  
   0311 864 WF1KPCH RETREG,759 ;\*\*\* for debugging... ;WAITFOR INTERRUPT  
   031B 865 IOFORK ;RETURN FROM ISR-  
   0065 31 0321 866 BRW RETREG ;CREATE FORK PROCESS (GJSB BACK TO ISR)  
   0321 867 ;

0324 869 : TRANSFER FUNCTION EXECUTION  
 0324 870 : FUNCTIONS INCLUDE:  
 0324 871 :  
 0324 872 :  
 0324 873 :  
 0324 874 : WRITE DATA  
 0324 875 : READ DATA  
 0324 876 :  
 0324 877 : INPUTS:  
 0324 878 R4 - DEVICE CSR ADDRESS  
 0324 879 R5 - UCB ADDRESS  
 0324 880 :  
 0324 881 : FUNCTIONAL DESCRIPTION:  
 0324 882 :  
 0324 883 : THE TRANSFER PARAMETERS ARE LOADED INTO THE CONSOLE REGISTER.  
 0324 884 : INTERRUPTS ARE LOCKED OUT, THE FUNCTION IS INITIATED, AND  
 0324 885 : A WAITFOR INTERRUPT AND KEEP CHANNEL IS EXECUTED.  
 0324 886 :  
 0324 887 : UPON RETURN FROM THE INTERRUPT SERVICE ROUTINE, THE TRANSFER WILL  
 0324 888 : EITHER BE COMPLETE OR AN ERROR WILL HAVE BEEN DETECTED.  
 0324 889 :  
 0324 890 :  
 0324 891 XFER: ;TRANSFER FUNCTION EXECUTION  
 0324 892 :  
 0324 893 : EXECUTE THE TRANSFER FUNCTION  
 0324 894 :  
 0324 895 DSBINT  
 06 64 A5 05 E1 032A BBC #UCBSV\_POWER,UCBSW\_STS(R5),20\$ ;BRANCH IF NOT POWERFAIL  
 0054 31 032F ENBINT  
 0332 BRW RETREG  
 0335 :  
 0335 896 : SET UP CONTENTS OF STXCS, AND SET MOVE ROUTINE ADDRESS  
 0335 897 : FOR USE IN INTERRUPT ROUTINE.  
 0335 898 :  
 0335 899 :  
 0335 900 :  
 0335 901 :  
 0335 902 :  
 50 53 46 8F 9A 0335 903 20\$: MOVZBL #<READ BLOCK!STXCS\_M\_IE>,R3 ;ASSUME READING  
 0B 00000000 GF 9E 0339 904 MOVAB G^IOCSMOVTOUSER,RO :SET MOVE ROUTINE ADDRESS  
 0B 00DD C5 00 E0 0340 905 BBS #CV V RD UCBSB CV STS(R5),40\$ ;BRANCH IF READING  
 50 53 45 8F 9A 0346 906 MOVZBL #<WRITE BLOCK!STXCS\_M\_IE>,R3 ;SET FOR WRITING  
 50 00000000 GF 9E 034A 907 MOVAB G^IOCSMOVFRUSER,RO ;SET MOVE ROUTINE ADDRESS  
 00DE C5 0100 8F B0 0351 908 40\$: MOVW #256,UCBSW CV BBC(R5) ;SET WORD COUNT  
 00E4 C5 50 D0 0358 909 MOVL RO,UCBSL\_CV\_MRTN(R5) ;SAVE MOVE ROUTINE ADDRESS  
 035D 910 :  
 035D 911 : SET LBN INTO R3  
 035D 912 :  
 53 10 08 00BC C5 F0 035D 913 INSV UCBSL\_MEDIA(R5),#STXCS\_V\_ADDRS,#STXCS\_S\_ADDRS,R3 :  
 50 50 7E A5 3C 0364 914 MOVZWL UCBSW\_BCNT(R5),RO :GET BYTE COUNT OF TRANSFER  
 50 00000200 8F C6 0368 915 DIVL2 #512,RO :COMPUTE # BLOCKS  
 0000004C BF 55 CO 036F 916 ADDL2 #2,RO :THROW IN 2 EXTRA FOR GOOD LUCK  
 0372 MTPR R3,#PR8\_STXCS :READ/WRITE REQUEST  
 0379 917 : ISR WILL NOT RETURN UNTIL COMPLETE  
 0379 918 : TRANSFER DONE OR ERROR DETECTED.  
 0379 919 WFIKPCH RETREG,RO :WAITFOR INTERRUPT AND KEEP CHANNEL  
 0383 920 : RETURN HERE FROM ISR SAVING REGISTERS  
 0383 921 IOFORK :CREATE FORK PROCESS (RETURN TO ISR)  
 0389 922 : RETURN HERE FROM ISR REI ROUTINE  
 0389 923 :  
 0389 924 :  
 0389 925 : GET STATUS AND RESET ERRORS

0389 926 :  
 0389 927 RETREG: ;GET STATUS AND RESET ERRORS  
 0389 928 :  
 0389 929 : DETERMINE EXIT - SPECIAL CONDITION, FATAL ERROR, RETRIABLE ERROR, OR SUCCESS  
 0389 930 :  
 7D 00DD CS 02 E4 0380 931 SETIPL UCBSB\_FIPL(R5) ;ENSURE AT FORK IPL  
 73 00DD CS 03 E4 0393 932 BBSC #CV\_V-STSError,UCBSB\_CV\_STS(R5),260\$ ;BRANCH IF GETSTS ERROR  
 02 00DC CS 91 12 039E 933 BBSC #CV\_V-ABORT,UCBSB\_CV\_STS(R5),240\$ ;BRANCH IF ISR SAID TO ABORT  
 50 00CC CS 7D 03A0 934 CMPB UCBSB\_CV\_STATE(R5),#ITC\_STS2 ;DID WE GET STATUS?  
 1D 51 05 00 ED 03A5 935 BNEQ 20\$ ;NO MUST BE POWERFAIL OR TIMEOUT  
 03AA 936 MOVO UCBSL\_CV\_CS(R5),R0 ;GET CS AND MP REGISTERS IN R0/R1  
 03AA 937 CMPZV #0,#5,R1,- ;HEADS, BRUSHES, STATE OK?  
 03AA 938 #<CV\_MP\_M\_BH!CV\_MP\_M\_HO!CV\_SLM> :  
 64 A5 0040 0E 13 03AA 939 BEQL 20\$ ;IF EQL - YES, ONLINE  
 50 01A4 8F AA 03AC 940 BICW2 #UCBSM\_TIMEOUT,UCBSW\_STS(R5) ;CLEAR DEVICE TIME OUT  
 FEDB 31 03B2 941 MOVZWL #SSS\_MEDOFL,R0 ;SET MEDIUM OFFLINE STATUS  
 64 A5 0060 8F B3 03BA 942 BRW FUNCXT ;RETURN  
 03 13 03C0 943 20\$: BITW #UCBSM\_POWER!- ;POWER FAIL OR DEVICE TIMEOUT?  
 004F 31 03C2 944 BEQL 30\$ ;IF EQL NO  
 03C5 945 BRW SPECOND ;YES - SPECIAL CONDITION  
 3E 51 09 E0 03C5 946 :  
 32 50 0F E1 03C9 947 :  
 00000000 GF 16 03CD 948 30\$: BBS #CV\_MP\_V\_VC,R1,200\$ ;IF SET - VOLUME INVALID  
 2E 009A CS 0F E0 03D3 949 40\$: BBC #CV\_CS\_V\_CE,R0,100\$ ;IF CLEAR RL11 OK  
 50 00CC CS 7D 03D9 950 40\$: JSB G^ERLSDEVICERR ;ALLOCATE AND FILL ERROR MESSAGE BUFFER  
 25 50 0D E0 03DE 951 BBS #IOSV\_INHRETRY,UCBSW\_FUNC(R5),200\$ ;IF SET - RETRY INHIBITED  
 OF 50 0E E1 03E2 952 MOVO UCBSL\_CV\_CS(R5),R0 ;GET CS AND MP REGISTERS IN R0/R1  
 04 51 0D E1 03E6 953 BBS #CV\_CS\_V\_NXM,R0,200\$ ;IF SET - NONEXISTENT MEMORY  
 19 51 0A E0 03EA 954 BBC #CV\_CS\_V\_DE,R0,80\$ ;IF CLR - NO DRIVE ERRORS  
 51 C500 8F B3 03EE 955 BBC #CV\_MP\_V\_WL,R1,60\$ ;IF CLR - NOT WRITE LOCKED  
 03F3 956 BBS #CV\_MP\_V\_WGE,R1,200\$ ;IF WL & WGE SET - WL ERROR  
 03F3 957 60\$: BITW #<CV\_MP\_M\_WDE!- ;WRITE DATA ERROR, OR  
 03F3 958 CV\_MP\_M\_CRE!- ;CURRENT HEAD ERROR, OR  
 03F3 959 CV\_MP\_M\_WGE!- ;WRITE GATE ERROR, OR  
 03F3 960 CV\_MP\_M\_DSE>,R1 ;DRIVE SELECT ERROR?  
 12 12 03F3 961 BNEQ 20\$ ;IF NEQ - YES  
 03F5 962 :  
 03F5 963 : RETRIABLE ERROR EXIT  
 7E 009C DS 98 03F5 964 :  
 009C CS 8E C0 03FA 965 80\$: CVTBL @UCBSL\_DPC(R5),-(SP) ;GET BRANCH DISPLACEMENT  
 03FF 966 ADDL2 (SP)+,UCBSL\_DPC(R5) ;CALCULATE RETURN ADDRESS - 1  
 03FF 967 :  
 03FF 968 : SUCCESSFUL OPERATION EXIT  
 03FF 969 :  
 009C CS D6 03FF 970 100\$: INCL UCBSL\_DPC(R5) ;ADJUST TO CORRECT RETURN ADDRESS  
 009C DS 17 0403 971 JMP UCBSL\_DPC(R5) ;RETURN TO DRIVER  
 0407 972 :  
 0407 973 : FATAL ERROR EXIT  
 0407 974 :  
 FE4F 10 0407 975 200\$: BSBP ABORT\_RESET\_STATUS ;DO AN ABORT AND RESET STATUS  
 31 0409 976 BRW FATALERR ;FATAL ERROR EXIT  
 040C 977 :  
 040C 978 : ISR DETECTED ERROR. TELL CONSOLE TO ABORT, AND TRY AGAIN IF WE CAN  
 040C 979 :  
 040C 980 :  
 55 10 040C 981 240\$: BSBP ABORT\_RESET\_STATUS ;ABORT AND RESET STATUS  
 E5 11 040E 982 BRB 80\$ ;TRY AGAIN IF RETRIES LEFT

0410 983 :  
 0410 984 : CONSOLE REPORTED ERROR DURING GET STATUS INTERRUPT  
 0410 985 :  
 4A 10 0410 986 260\$: BSB B RESET\_STATUS\_ONLY :RESET STATUS ONLY  
 E1 11 0412 987 BRB 80\$ ;TRY AGAIN IF RETRIES LEFT  
 0414 988 :  
 0414 989 : SPECIAL CONDITION EXIT (POWER FAILURE OR DEVICE TIMEOUT)  
 0414 990 :  
 0414 991 SPECOND:  
 2B 64 A5 05 E0 0414 992 BBS #UCBSV\_POWER,UCBSW\_STS(R5),PWRFAIL ;IF SET - POWER FAILURE  
 0419 993 ;IF CLR - DEVICE TIMEOUT  
 64 A5 00000000'GF 16 0419 994 JSB G^ERLSDEVICTMO ;LOG DEVICE TIMEOUT  
 50 022C 8F AA 041F 995 BICW2 #UCBSM\_TIMEOUT,UCBSW\_STS(R5) ;CLEAR TIMEOUT STATUS  
 0080 C5 97 0425 996 MOVZWL #SSS\_TIMEOUT,R0 ;SET DEVICE TIMEOUT STATUS  
 05 13 042A 997 DECB UCBSB\_ERTCNT(R5) ;ANY ERROR RETRIES REMAINING?  
 31 10 0430 998 BEQL RESETXFR ;IF EQL - NO  
 FD97 31 0432 999 BSBB ABORT\_RESET\_STATUS ;ABORT AND RESET STATUS  
 0433 1000 BRW FD DISPATCH ;RETRY FUNCTION AGAIN  
 0435 1001 :  
 0435 1002 RESETXFR:  
 00C0 53 2C 10 0435 1003 BSB ABORT\_RESET\_STATUS ;RESET TRANSFER BYTE COUNT  
 58 A5 DD 0437 1004 MOVL UCBSL\_IRP(R5),R3 ;ABORT TRANSFER AND RESET STATUS  
 32 A3 AE 043B 1005 MNEGW IRPSL\_BCNT(R3),UCBSW\_BCR(R5) ;GET ADDRESS OF I/O PACKET  
 FES1 31 0441 1006 BRW FUNCXT ;RESET BYTE COUNT  
 0444 1007 ;EXIT  
 64 A5 20 AA 0444 1008 PWRFAIL: ;POWER FAILURE  
 19 10 0448 1009 BICW2 #UCBSM\_POWER,UCBSW\_STS(R5) ;CLEAR POWER FAILURE BIT  
 044A 1010 BSB ABORT\_RESET\_STATUS ;ABORT AND RESET STATUS  
 53 58 A5 00 0450 1011 RELCHAN ;RELEASE CHANNEL IF OWNED  
 2C A3 7D 0454 1012 MOVL UCBSL\_IRP(R5),R3 ;GET ADDRESS OF I/O PACKET  
 78 A5 0457 1013 MOVO IRPSL\_SVAPTE(R3),- ;RESTORE TRANSFER PARAMETERS  
 FDOF 31 0459 1014 UCBSL\_SVAPTE(R5) ;  
 045C 1015 BRW PREPROCESS ;RETURN TO PREPROCESS UCB FIELDS  
 045C 1016 :  
 045C 1017 : ISSUE AN ABORT TO THE CONSOLE. WHEN THE ABORT COMPLETES, READ  
 045C 1018 : THE RL11 STATUS REGISTERS, ASSERTING RST.  
 045C 1019 :  
 045C 1020 : THIS ROUTINE DESTROYS R0-R3  
 045C 1021 :  
 045C 1022 : .ENABLE LOCAL\_BLOCK  
 045C 1023 :  
 045C 1024 RESET\_STATUS\_ONLY:  
 00F8 C5 BED0 045C 1025 POPL UCBSL\_CV\_ABPC(R5) ;POP RETURN ADDRESS FROM STACK  
 28 11 0461 1026 BRB 30\$ ;GO EXECUTE  
 0463 1027 :  
 0463 1028 ABORT\_RESET\_STATUS:  
 00DC 00F8 C5 BED0 0463 1029 POPC UCBSL\_CV\_ABPC(R5) ;POP RETURN ADDRESS FROM STACK  
 03 90 0468 1030 MOVB #ITC\_ABORT,UCBSB\_CV\_STATE(R5) ;SET DISPATCH  
 046D 1031 DSBINT ;DISABLE INTERRUPTS  
 50 63 8F 9A 0473 1032 MOVZBL #ABORT\_TRANSFER!STXCS\_M IE>,R0 ;SETUP FUNCTION  
 0000004C 8F 50 DA 0477 1033 MTPR R0,#PRS\_STXCS ;TELL THE CONSOLE TO ABORT  
 047E 1034 WF IKPCH 20\$,#6  
 0488 1035 20\$: IOFORK  
 00D4 C5 00CC C5 7D 048E 1036 30\$: MOVO UCBSL\_CV\_CS(R5),UCBSQ\_CV\_CSMP(R5) ;SAVE CS/MP REGISTERS  
 0495 1037 TIMEDWAIT TIME=#600\*1000,- ;WAIT FOR CONSOLE TO BE READY  
 0495 1038 INS1=<MFPR #PRS\_STXCS,R2>,- ;READ STATUS REGISTER  
 0495 1039 INS2=<BBS #STXCS\_V\_RDY,R2,40\$>,- ;BRANCH IF READY

0495 1040 : DONELBL=40\$ ;TO SAME PLACE AS DONE  
04C2 1041 :  
04C2 1042 : \*\* WHAT DO WE DO IF THE CONSOLE DOES NOT GO READY IN TIME?  
04C2 1043 :  
0000004C BF 00000042 BF DA 04C2 1044 : MOV#ITC\_RESET1,UCBSB\_CV\_STATE(R5) ;SET DISPATCH  
0000004C BF 00000042 BF DA 04C7 1045 : DSBINT ;DISABLE INTERRUPTS  
04CD 1046 : MTPR #<STATUS\_RESET!STXCS\_M\_I>,#PRS STXCS ;REQUEST STATUS WITH RST ASSE  
04D8 1047 : WFIKPCH 60\$,#6 ;WAITFOR INTERRUPT  
00CC C5 00D4 C5 7D 04E2 1048 60\$: IOFORK  
00F8 D5 17 04EF 1049 : MOVQ UCBSQ\_CV\_CSMP(R5),UCBSL\_CV\_CS(R5) ;RESTORE CS/MP REGISTERS  
04F3 1050 : JMP @UCBSL\_CV\_ABPC(R5) ;RETURN TO CALLER  
04F3 1051 :  
04F3 1052 : .DISABLE LOCAL\_BLOCK

04F3 1054 .SBTTL INTERRUPT SERVICE ROUTINE  
 04F3 1055 :++  
 04F3 1056 : CV\$INT - VAX 8600 CONSOLE RL02 INTERRUPT SERVICE ROUTINE  
 04F3 1057 :  
 04F3 1058 : FUNCTIONAL DESCRIPTION:  
 04F3 1059 :  
 04F3 1060 : THIS ROUTINE IS ENTERED VIA A JSB INSTRUCTION WHEN AN INTERRUPT  
 04F3 1061 : OCCURS ON THE VAX 8600 CONSOLE STXCS REGISTER. IF THE INTERRUPT  
 04F3 1062 : IS NOT EXPECTED, THE UNSOLICITED INTERRUPT ROUTINE DISMISSES  
 04F3 1063 : THE INTERRUPT. IF THE INTERRUPT IS EXPECTED, DEVICE REGISTERS  
 04F3 1064 : ARE SAVED AND THE DRIVER IS CALLED AT ITS INTERRUPT RETURN ADDRESS.  
 04F3 1065 : THE DRIVER FORKS, CAUSING A RETURN TO THIS ROUTINE,  
 04F3 1066 : WHICH RESTORES GENERAL REGISTERS AND DISMISSES THE INTERRUPT.  
 04F3 1067 :  
 04F3 1068 :  
 04F3 1069 :  
 04F3 1070 : 00(SP) - POINTER TO ADDRESS OF THE IDB  
 04F3 1071 : 04(SP) - SAVED R0  
 04F3 1072 : 08(SP) - SAVED R1  
 04F3 1073 : 12(SP) - SAVED R2  
 04F3 1074 : 16(SP) - SAVED R3  
 04F3 1075 : 20(SP) - SAVED R4  
 04F3 1076 : 24(SP) - SAVED R5  
 04F3 1077 : 28(SP) - PC AT THE TIME OF THE INTERRUPT  
 04F3 1078 : 32(SP) - PSL AT THE TIME OF THE INTERRUPT  
 04F3 1079 :  
 04F3 1080 :  
 04F3 1081 :  
 04F3 1082 : DEVICE REGISTERS ARE SAVED, IPL IS LOWERED TO FORK LEVEL, THE  
 04F3 1083 : INTERRUPT IS DISMISSED, ALL REGISTERS EXCEPT R0-R5 ARE PRESERVED.  
 04F3 1084 :  
 04F3 1085 :--  
 04F3 1086 :  
 04F3 1087 CV\_INT: :  
 53 9E D0 04F3 1088 MOVL @(SP)+,R3 : INTERRUPT SERVICE ROUTINE  
 54 63 7D 04F6 1089 MOVQ (R3),R4 : REMOVE ADDRESS OF IDB FROM STACK  
 55 D5 04F9 1090 TSTL R5 : GET ADDRESS OF CSR AND UCB  
 27 13 04FB 1091 BEQL CV\_UNSOLNT : IS R5 A ZERO  
 53 0000004C 8F DB 04FD 1092 #PRS\_STXCS,R3 : IF EQL NO OWNER  
 1C 53 07 E1 0504 1093 BBC #STXCS\_V\_RDY,R3,CV\_UNSOLNT : \*\*TEMP\*\* READ CONSOLE STATUS  
 01 E5 0508 1094 BBCC #UCBSV\_INT- : \*\*TEMP\*\* BRANCH IF NOT READY  
 17 64 AS 050A 1095 UCBSW\_STS(R5),CV\_UNSOLNT : IF CLR - INTERRUPT NOT EXPECTED  
 53 00DC C5 9A 0500 1096 CV\_INT\_DISP: :  
 40 13 0502 1097 MOVZBL UCBSB CV\_STATE(R5),R3 : GET INTERRUPT STATE  
 0512 1098 BEQL CV\_INT\_XFR : BRANCH IF TRANSFER INTERRUPT  
 0514 1099 CASE R3,<- : AND DISPATCH  
 0514 1100 CV\_INT\_XFR- : TRANSFER INTERRUPT  
 0514 1101 CV\_INT\_STS1,- : FIRST PART OF STATUS  
 0514 1102 CV\_INT\_STS2,- : SECOND PART OF STATUS  
 0514 1103 CV\_INT\_ABORT,- : ABORT REQUEST  
 0514 1104 CV\_INT\_RSTS1,- : GET STATUS WITH RST ASSERTED  
 0514 1105 CV\_INT\_RSTS2>,-  
 0514 1106 TYPE=B  
 0524 1107 :  
 3F BA 0524 1108 CV\_UNSOLNT: :  
 02 0526 1109 POPR #^M<R0,R1,R2,R3,R4,R5> : UNSOLICITED INTERRUPT  
 1110 REI : RESTORE R0-R5  
 : RETURN FROM INTERRUPT

00CC C5 0000004D 8F DB 0527 1111 : GET STATUS WITH RESET INTERRUPT

0000004C 8F 00DC C5 05 90 0527 1112 : CV\_INT\_RSTS1:

64 A5 02 08 0530 1113 MFPR #PRS\_STXDB,UCBSL\_CV\_CS(R5) ;READ CONTROL/STATUS REGISTER

DE 11 0535 1114 MOVB #ITC\_RESET2,UCBSB\_CV\_STATE(R5) ;SET NEXT STATE

0000004D 8F 00000042 8F DA 0540 1115 MTPR #<STATUS RESET!STXCS\_M IE>,#PRS\_STXCS

OC B5 02 08 0544 1116 BISB2 #UCBSM\_INT,UCBSU\_STS(R5) ;FLAG INTERRUPT EXPECTED

DO 11 0546 1117 BRB CV\_UNSLNT

0000 C5 0000004D 8F DB 0546 1118 : CV\_INT\_RSTS2:

OC B5 16 054F 1119 MFPR #PRS\_STXDB,UCBSL\_CV\_MP(R5) ;SAVE MULTIPURPOSE REGISTER

DO 11 0552 1120 JSB UCBSL\_FPC(R5) :CALL DRIVER AT INTERRUPT RETURN ADDR

0000 C5 0000004D 8F DB 0554 1121 BRB CV\_UNSLNT

OC B5 16 0554 1122 : TRANSFER INTERRUPT

DO 11 0554 1123 : CV\_INT\_XFR:

53 0000004C 8F DB 0554 1124 MFPR #PRS\_STXCS\_R3 ;GET STATUS REGISTER

53 08 18 EF 0558 1125 EXTZV #STXCS\_V\_STS,#STXCS\_S\_STS,R3,R3 ;GET CONSOLE RL02 STATUS

53 02 D1 0560 1126 CMPL #TRANS\_CONTINUE,R3 :CONTINUE TRANSACTION?

03 13 0563 1127 BEQL 10\$ :YES, CONTINUE

64 00DD C5 00 E1 0565 1128 BRW 400\$ :ELSE, BRANCH TO ABORT

00CE 31 0568 1129 BBC #CV\_V\_RD,UCBSB\_CV\_STS(R5),200\$ ;BRANCH IF WRITING

0000004C 8F DB 056E 1130 : OPERATION IS A READ FROM DISK

51 10 08 00BC C5 F0 056E 1131 : 10\$:

0000004C 8F S1 DA 056E 1132 MFPR #PRS\_STXDB,-(SP) ;READ DATA ONTO STACK

0000 00 C5 B5 0575 1133 MOVZBL #<READ BLOCK!STXCS\_M IE>,R1 ;SET NEXT READ

22 13 0580 1134 INSV UCBSDL\_MEDIA(R5),#STXCS\_V\_ADDRS,#STXCS\_S\_ADDRS,R1 ;SET LBN

S1 5E DO 0587 1135 MTPR R1,#PRS\_STXCS ;CONTINUE READING

S2 02 DO 0588 1136 TSTW UCBBSW\_BCR(R5) ;HAVE WE COMPLETED THE REQUEST?

50 00E8 C5 DO 0589 1137 BEQL 20\$ :IF EQL YES, DON'T WRITE TO BUFFER

00E4 D5 16 0590 1138 MOVL SP,R1 :GET ADDRESS OF DATA

00E8 C5 50 DO 0593 1139 MOVL #2,R2 :WRITE 2 BYTES INTO USER BUFFER

00000000 GF 9E 05A1 1140 MOVAB UCBSDL\_CV\_BUFWIN(R5),R0 :GET BUFFER ADDRESS

0000 02 A0 05AA 1141 ADDW2 #UCBSL\_CV\_MVRTN(R5) :WRITE INTO USER BUFFER

SE 06 C0 05AF 1142 JSB R0,UCBSL\_CV\_BUFWIN(R5) :SAVE WINDOW INTO USER BUFFER

00DE C5 B7 05B2 1143 MOVL G:[0CSMOVTOUSER2,UCBSL\_CV\_MVRTN(R5)] ;SET MOVE ROUTINE ADDRESS

12 19 05B6 1144 ADDL2 #2,UCBSW\_BCR(R5) :COUNT TWO MORE BYTES TRANSFERRED

54 0000004C 8F DB 05B8 1145 DECW #4,SP :CLEAR DATA FROM STACK

91 54 07 F0 05BF 1146 BLSS UCBBSW\_CV\_BBC(R5) :COUNT ANOTHER WORD TRANSFERRED

64 A5 02 08 05C3 1147 BBS #STXCS\_V\_RDY,R4,CV\_INT\_XFR :PROTOCOL ERROR

00DD C5 FF5A 31 05C7 1148 BISB2 #UCBSM\_INT,UCBSU\_STS(R5) ;BRANCH IF DONE AGAIN

08 88 05CA 1149 BRW CV\_UNSLNT ;FLAG INTERRUPT EXPECTED

FF7D 31 05CF 1150 100\$: BRW #CV\_M\_ABORT,UCBSB\_CV\_STS(R5) ;EXIT THIS INTERRUPT

0000 C5 B5 05D2 1151 120\$: BRW #CV\_INT\_ABORT ;FLAG TO ABORT AND RETRY

0000 02 13 05D6 1152 : CALL DRIVER TO DO IT

0000 C5 B5 05D2 1153 1160 : WRITING TO DISK

0000 24 13 05D6 1154 : 1161 : REQUEST COMPLETE?

S1 00E0 C5 9F 05D8 1155 BEQL 200\$: TSTW UCBBSW\_BCR(R5) :IF EQL YES DON'T BOTHER FETCHING

52 02 DO 05DD 1156 MOVAB UCBSDL\_CV\_IBUF(R5),R1 :GET ADDRESS OF INTERNAL BUFFER

50 00E8 C5 DO 05E0 1157 1162 : MOVL #2,R2 :SET NUMBER OF BYTES

1167 : MOVL UCBSDL\_CV\_BUFWIN(R5),R0 :GET BUFFER WINDOW

00E4 DS 16 0SE5 1168 JSB #UCBSL\_CV\_MVRTN(R5) :GET 2 BYTES FROM USERS BUFFER  
 00E4 C5 50 D0 0SE9 1169 MOVL R0\_UCBSL\_CV\_BUFWIN(R5) :SAVE WINDOW  
 00E8 C5 50 9E 0SEE 1170 MOVAL G\$OC\$MOVFRSER2,UCBSL\_CV\_MVRTN(R5) :SET MOVE ROUTINE  
 00C0 C5 02 A0 0SF7 1171 ADDW2 #2\_UCBSW\_BCR(R5) :COUNT TWO MORE BYTES  
 0000004D BF DA 05FC 1172 2208: MTPR UCBSL\_CV\_IBUF(R5),#PRS\_STXDB :WRITE WORD TO CONSOLE  
 00C0 C5 B5 0605 1173 TSTW UCBSW\_BCR(R5) :REQUEST COMPLETE?  
 00C0 C5 B5 0609 1174 BNEQ 2408 :IF NEQ NO  
 00E0 C5 D4 060B 1175 CLRL UCBSL\_CV\_IBUF(R5) :YES, CLEAR BUFFER SO WE WRITE 0'S  
 00DE C5 B7 060F 1176 2408: DECW UCBSW\_CVBBC(R5) :COUNT ANOTHER WORD TRANSFERRED  
 00E0 C5 B7 19 19 0613 1177 2508: BLSS 2808 :PROTOCOL ERROR  
 00E0 C5 B7 19 19 0613 1178 MOVZBL #<WRITE\_BLOCK!STXCS M IE>,R1 :REQUEST TO SEND AGAIN  
 S1 10 08 00BC C5 F0 0619 1179 INSV UCBSL MEDIA(R5),#STXCS\_V\_ADDR\$ #STXCS\_S\_ADDR\$,R1 :SET LBN  
 0000004C BF 51 DA 0620 1180 MTPR R1,#PRS\_STXCS :SEND COMMAND TO CONSOLE  
 64 AS 02 88 0627 1181 2608: BISB2 #UCBSM\_INT,UCBSW\_STS(R5) :FLAG INTERRUPT EXPECTED  
 00DD C5 08 FECB 31 062E 1182 BRW CV\_UNSLNT :DISMISS INTERRUPT  
 00DD C5 08 FF19 31 0633 1183 2808: BISB2 #CV\_M\_ABORT,UCBSB\_CV\_STS(R5) :FLAG TO ABORT AND RETRY  
 00DD C5 08 FF19 31 0633 1184 BRW CV\_INT\_ABORT :CALL DRIVER TO DO IT  
 00DD C5 08 FF19 31 0636 1185 :  
 00DD C5 08 FF19 31 0636 1186 : TRANSACTION COMPLETE, OR ERROR DETECTED. REQUEST STATUS  
 00DD C5 08 FF19 31 0636 1187 :  
 53 80 8F 91 0636 1188 4008: CMPB #HANDSHAKE\_ERROR,R3 :WAS THERE A HANDSHAKE ERROR?  
 1D 13 063A 1189 BEQL 4408 :BRANCH IF YES  
 00DE C5 B5 063C 1190 TSTW UCBSW\_CVBBC(R5) :ALL WORDS TRANSFERRED?  
 18 12 0640 1191 BNEQ 4608 :IF NEQ NO  
 00DC C5 01 90 0642 1192 4208: MOVB #ITC\_STS1,UCBSB\_CV\_STATE(R5) :SET NEXT STATE  
 00000044 BF DA 0647 1193 MTPR #<READ\_STATUS!STXCS M IE>,#PRS\_STXCS :REQUEST STATUS  
 64 AS 02 88 0652 1194 BISB2 #UCBSM\_INT,UCBSW\_STS(R5) :FLAG INTERRUPT EXPECTED  
 FECB 31 0656 1195 BRW CV\_UNSLNT :DISMISS INTERRUPT  
 00DD C5 08 FEED 31 0659 1196 :  
 00DD C5 08 FEED 31 0659 1197 : HANDSHAKE ERROR. TELL DRIVER TO ABORT AND RETRY  
 01 0659 1198 4408: NOP :\*\*DEBUG  
 065A 1200 :  
 065A 1201 : NOT ALL WORDS TRANSFERRED. TELL DRIVER TO ABORT AND RETRY  
 065A 1202 :  
 00DD C5 08 FEED 31 065A 1203 4608: BISB2 #CV\_M\_ABORT,UCBSB\_CV\_STS(R5) :FLAG TO ABORT AND RETRY  
 065F 1204 BRW CV\_INT\_ABORT :CALL DRIVER TO DO IT  
 0662 1205 :  
 0662 1206 : ERROR ON GET STATUS OPERATION  
 0662 1207 :  
 00DD C5 06 FEED 31 0662 1208 CV\_STSERROR:  
 0662 1209 BISB2 #CV\_M\_STSERROR,UCBSB\_CV\_STS(R5) :FLAG GET STATUS ERROR  
 0667 1210 BRW CV\_INT\_ABORT :CALL DRIVER TO PROCESS ERROR  
 066A 1211 :  
 066A 1212 : GET STATUS PART 1 INTERRUPT  
 066A 1213 :  
 53 0000004C BF DB 066A 1214 CV\_INT\_STS1:  
 ED 53 1F E0 0671 1215 MFPR #PRS\_STXCS,R3 :READ STXCS REGISTER  
 00CC C5 0000004D 3F DB 0675 1216 BBS #31\_R3,CV\_STSERROR :BRANCH IF ERROR GETTING STATUS  
 00DC C5 02 90 067E 1217 MFPR #PRS\_STXDB,UCBSL\_CV\_CS(R5) :GET THE CONTROL/STATUS REGISTER  
 0000004C BF DA 0683 1218 MOVB #ITC\_STS2,UCBSB\_CV\_STATE(R5) :SET NEXT STATE  
 64 AS 02 88 068E 1219 MTPR #<READ\_STATUS!STXCS M IE>,#PRS\_STXCS :REQUEST IT  
 FECB 31 0692 1220 BISB2 #UCBSM\_INT,UCBSW\_STS(R5) :FLAG INTERRUPT EXPECTED  
 0695 1221 BRW CV\_UNSLNT :DISMISS INTERRUPT  
 0695 1222 :  
 0695 1223 : GET STATUS PART 2 INTERRUPT  
 0695 1224 :

53 0000004C BF DB 0695 1225 CV\_INT\_STS2:  
 C2 55 1F EO 0695 1226 MFPR #PRS\_STXCS\_R3 ;READ STXCS REGISTER  
 0000 CS 0000004D BF DB 06A0 1227 BBS #31 R3.CV\_STSERROR ;BRANCH IF ERROR GETTING STATUS  
 24 0000 CS 01 EO 06A9 1228 MFPR #PRS\_SFDB\_UCBSL(CV\_MP(R5)) ;GET MULTIPURPOSE REGISTER  
 06AF 1229 BBS #CV\_V\_STSONLY\_UCBSB\_CV\_STS(R5),208 ;BRANCH IF STATUS ONLY  
 06AF 1230 : TRANSFER OF A BLOCK IS COMPLETE. SEE IF ERRORS, AND PROCESS IF SO.  
 06AF 1231 :  
 06AF 1232 : IF NO ERRORS, THEN SEE IF DONE WITH COMPLETE TRANSFER  
 06AF 1233 :  
 1D 50 00CC CS 7D 06AF 1234 MOVO UCBSL\_CV\_CS(R5),R0 ;GET CS AND MP REGISTERS  
 51 05 00 ED 0684 1235 CMPZV #0,#5,R1- ;HEADS AND BRUSHES OK?  
 0689 1236 :<CV\_MP\_M\_BH:CV\_MP\_M\_HO:CV\_SLM>  
 50 E000 BF B5 12 0689 1237 BNEQ 208 ;IF NEQ NO  
 11 12 06C0 1238 BITW #<CV\_CS\_M\_CE:CV\_CS\_M\_DE:CV\_CS\_M\_NXM>,R0  
 51 C700 BF B3 06C2 1240 BNEQ 208 ;IF NEQ ERROR OF SOME SORT  
 06C7 1241 BITW #<CV\_MP\_M\_WDE:-  
 06C7 1242 CV\_MP\_M\_CRE:-  
 06C7 1243 CV\_MP\_M\_WGE:-  
 06C7 1244 CV\_MP\_M\_DSE:-  
 0A 12 06C7 1245 BNEQ 208 CV\_MP\_M\_VC>,R1 ;ANY ERRORS?  
 00BC CS D6 06C9 1246 INCL UCBSL\_MEDIA(R5) ;IF NEQ YES  
 00C0 CS B5 06CD 1247 TSTW UCBSW\_BCR(R5) ;NEXT LBN  
 06 12 06D1 1248 BNEQ 40S ;ARE WE DONE YET?  
 OC B5 16 06D3 1249 208: JSB UCBSL\_FPC(R5) ;BRANCH IF NOT DONE YET  
 FE4B 31 06D6 1250 308: BRW CV\_UNSOLNT ;CALL DRIVER AT INTERRUPT RETURN ADDR  
 06D9 1251 :  
 06D9 1252 : DISMISS  
 06D9 1253 : MORE DATA TO TRANSFER STILL  
 000E CS 0100 BF B0 06D9 1254 40S: MOVW #256,UCBSW\_CV\_BBC(R5) ;RESET BYTE COUNT FOR BLOCK  
 000C CS 94 06E0 1255 CLR BUCBSB\_CV\_STATE(R5) ;RESET STATE TO READ MODE  
 53 46 BF 9A 06E4 1256 MOVZBL #<READ BLOCK!STXCS\_M\_IE>,R3 ;ASSUME READING  
 04 0000 CS 00 EO 06E8 1257 BBS #CV\_V\_RD\_UCBSB\_CV\_STS(R5),60\$ ;BRANCH IF READING  
 53 45 BF 9A 06EE 1258 MOVZBL #<WRITE BLOCK!STXCS\_M\_IE>,R3 ;SET FOR WRITING  
 53 10 08 00BC CS F0 06F2 1259 60S: INSV UCBSL\_MEDIA(R5),#STXCS\_V\_ADDRS,#STXCS\_S\_ADDRS,R3 ;SET LBN  
 0000004C BF 53 DA 06F9 1260 MTPR R3,#PRS\_STXCS ;SEND COMMAND TO CONSOLE  
 64 A5 02 88 0700 1261 BISB2 #UCBSM\_INT\_UCBSW\_STS(R5) ;FLAG INTERRUPT EXPECTED  
 FE1D 31 0704 1262 BRW CV\_UNSOLNT ;EXIT INTERRUPT  
 0707 1263 :  
 00000787 0707 1264 CVPATCH: :  
 .BLKL 32

0787 1267 .SBTTL REGISTER DUMP ROUTINE  
 0787 1268 :\*\*  
 0787 1269 : CV\_REGDUMP - REGISTER DUMP ROUTINE  
 0787 1270 : FUNCTIONAL DESCRIPTION:  
 0787 1271 : THIS ROUTINE IS CALLED TO SAVE THE DEVICE REGISTERS AND UBA RESOURCE  
 0787 1272 : REGISTERS IN A SPECIFIED BUFFER. IT IS CALLED FROM THE DEVICE ERROR  
 0787 1273 : LOGGING ROUTINE AND FROM THE DIAGNOSTIC BUFFER FILL ROUTINE.  
 0787 1274 :  
 0787 1275 :  
 0787 1276 :  
 0787 1277 :  
 0787 1278 :  
 0787 1279 :  
 0787 1280 :     R0     - ADDRESS OF REGISTER SAVE BUFFER  
 0787 1281 :     R4     - ADDRESS OF DEVICE CONTROL STATUS REGISTER (CSR)  
 0787 1282 :     R5     - ADDRESS OF UNIT CONTROL BLOCK (UCB)  
 0787 1283 :  
 0787 1284 :  
 0787 1285 :  
 0787 1286 :     THE DEVICE AND UBA REGISTERS ARE SAVED IN THE SPECIFIED BUFFER.  
 0787 1287 :     R0 CONTAINS THE ADDRESS OF THE NEXT EMPTY LONGWORD IN THE BUFFER.  
 0787 1288 :     ALL REGISTERS EXCEPT R1 AND R2 ARE PRESERVED.  
 0787 1289 :  
 0787 1290 :--  
 0787 1291 :  
 0787 1292 CV\_REGDUMP:  
 80 80 09 00 00CC C5 3C 0787 1293 MOVL #<CV\_NUM\_REGS+5>, (R0)+ ;REGISTER DUMP ROUTINE  
 80 80 09 00 00CC C5 3C 078A 1294 MOVZWL UCBSL\_CV\_CS(R5), (R0)+ ;INSERT NUMBER OF REGISTERS  
 80 80 7C 078F 1295 CLRQ (R0)+ ;COPY CONTROL/STATUS REGISTER  
 80 80 7C 0791 1296 MOVZWL UCBSL\_CV\_MP(R5), (R0)+ ;NO BA/DA REGISTERS  
 80 80 7C 0796 1297 CLRQ (R0)+ ;COPY MULTIPURPOSE REGISTER  
 80 80 7C 0798 1298 CLRQ (R0)+ ;NO DATAPATH NUMBER/DATAPATH REGISTER  
 80 80 D4 079A 1299 CLRQ (R0)+ ;NO FINAL MAP REG/PREVIOUS MAP REG  
 80 80 05 079C 1300 RSB ;NO DATAPATH PURGE ERROR REGISTER  
 079D 1301 :  
 079D 1302 CV\_END: .END ;RETURN  
 079D 1303 :  
 :ADDRESS OF LAST LOCATION IN DRIVER

SSS	= 00000020	R	02	CV_MP_M_CHE	= 00004000
SSGENF_CODE	= 00000012			CV_MP_M_DSE	= 0000100
SSOP	= 00000002			CV_MP_M_HO	= 0000010
ABORT_RESET STATUS	= 00000463	R	03	CV_MP_M_VC	= 0000200
ABORT_TRANSFER	= 00000003			CV_MP_M_WDE	= 00008000
ACPSACCESS	*****	X	03	CV_MP_M_WUE	= 0000400
ACPSDEACCESS	*****	X	03	CV_MP_V_VC	= 0000009
ACPSMODIFY	*****	X	03	CV_MP_V_WGE	= 000000A
ACPSMOUNT	*****	X	03	CV_MP_V_WL	= 000000D
ACPSREADBLK	*****	X	03	CV_M_ABORT	= 0000008
ACPSWRITEBLK	*****	X	03	CV_M_RD	= 0000001
ATS_UBA	= 00000001			CV_M_STSError	= 0000004
AVAILABLE	= 0000023C	R	03	CV_M_STSONLY	= 0000002
CDF_AVAILABLE	= 00000011			CV_NOM_REGS	= 0000004
CDF_DRVCLR	= 00000004			CV_REGDUMP	0000787 R 03
CDF_NOP	= 00000010			CV_RLOX_INIT	000000D9 R 03
CDF_OFFSET	= 00000006			CV_RL11_INIT	000000D8 R 03
CDF_PACKACK	= 00000008			CV_SLM	= 0000005
CDF_READDATA	= 0000000C			CV_STARTIO	000016B R 03
CDF_READHEAD	= 0000000F			CV_STSError	00000662 R 03
CDF_RECAL	= 00000003			CV_UNSOLNT	00000524 R 03
CDF_RELEASE	= 00000005			CV_V_ABORT	= 0000003
CDF_RETCENTER	= 00000007			CV_V_RD	= 0000000
CDF_SEARCH	= 00000009			CV_V_STSError	= 0000002
CDF_SEEK	= 00000002			CV_V_STSONLY	= 0000001
CUF_UNLOAD	= 00000001			DC8_DISK	= 0000001
CDF_WRITECHECK	= 0000000A			DBBSK_CART	= 0000002
CDF_Writedata	= 00000008			DBBSL_ACPD	= 0000010
CDF_WRITEHEAD	= 0000000D			DBBSL_DDT	= 000000C
CRBSL_INTD	= 00000024			DEVSM_AVL	= 0040000
CVSDDT	00000000	RG	03	DEVSM_DIR	= 0000008
CVC_GETSTS	0000010A	R	03	DEVSM_ELG	= 0040000
CVPATCH	00000707	RG	03	DEVSM_FOD	= 00004000
CV_ALIGN	0000015B	R	03	DEVSM_IDV	= 04000000
CV_CS	00000000			DEVSM_ODV	= 08000000
CV_CS_M_CE	= 00008000			DEVSM_RND	= 10000000
CV_CS_M_DE	= 00004000			DEVSM_SHR	= 00010000
CV_CS_M_NXM	= 00002000			DO_FUNCTION	000022A R 03
CV_CS_V_CE	= 0000000F			DPTSC_LENGTH	= 0000038
CV_CS_V_CRC	= 00000008			DPTSC_VERSION	= 0000004
CV_CS_V_CVT	= 0000000C			DPTSINITAB	0000038 R 02
CV_CS_V_DE	= 0000000E			DPTSM_SVP	= 0000002
CV_CS_V_DRDY	= 00000000			DPTSREINITAB	0000067 R 02
CV_CS_V_NXM	= 0000000D			DPTSTAB	0000000 R 02
CV_CS_V_OPI	= 0000000A			DRVCLR	000022A R 03
CV_END	0000079D	R	03	DTS_RL02	= 000000A
CV_FUNCTABLE	00000038	R	03	DYNSC_CRB	= 0000005
CV_INT	000004F3	RG	03	DYNSC_DDB	= 0000006
CV_INT_ABORT	0000054F	R	03	DYNSC_DPT	= 0000001E
CV_INT_DISP	00000500	R	03	DYNSC_UCB	= 00000010
CV_INT_RSTS1	00000527	R	03	EMBSL_DV_REGSAR	= 000004E
CV_INT_RSTS2	00000546	R	03	ERLSDEVICERR	***** X 03
CV_INT_STS1	0000066A	R	03	ERLSDEVICETMO	***** X 03
CV_INT_STS2	00000695	R	03	EXESABORTIO	***** X 03
CV_INT_XFR	00000554	R	03	EXESGL_TENUSEC	***** X 03
CV_MP	00000002			EXESGL_UBDELAY	***** X 03
CV_MP_M_BH	= 00000008			EXESIOPRK	***** X 03

EXESLCLDSKVALID	*****	X	03	IRPSW_BCNT	= 00000032
EXESONEPARM	*****	X	03	IRPSW_FUNC	= 00000020
EXESSENSEMODE	*****	X	03	IRPSW_STS	= 0000002A
EXESSETCHAR	*****	X	03	ITC_ABORT	= 00000003
EXESZEROPARM	*****	X	03	ITC_DATA	= 00000000
FATALERR	0000025B	R	03	ITC_RESET1	= 00000004
FDISPATCH	000001CC	R	03	ITC_RESET2	= 00000005
FEXL	000002CB	R	03	ITC_STS1	= 00000001
FUNCTAB_LEN	= 000000A0			ITC_STS2	= 00000002
FUNCXT	= 00000295	R	03	MASKH	= 00000008
HANDSHAKE_ERRUR	= 00000080			MASKL	= 04000000
HW_ERROR	= 00000081			NOP	0000022A R 03
IDBSL_CSR	= 00000000			NORMAL	0000024D R 03
IDBSL_OWNER	= 00000004			NO_OP	= 00000000
IMMED	000002EB	R	03	PACKACK	00000234 R 03
IOSV_INHRETRY	= 0000000F			PRS_IPL	= 00000012
IOS_ACCESS	= 00000032			PRS_STXCS	= 0000004C
IOS_ACPCONTROL	= 00000038			PRS_STXDB	= 0000004D
IOS_AVAILABLE	= 00000011			PREPROCESS	00000168 R 03
IOS_CREATE	= 00000033			PWRFAIL	00000444 R 03
IOS_DEACCESS	= 00000034			READDATA	00000244 R 03
IOS_DELETE	= 00000035			READ_BLOCK	= 00000006
IOS_DRVCLR	= 00000004			READ_STATUS	= 00000004
IOS MODIFY	= 00000036			RESETXFR	00000435 R 03
IOS_MOUNT	= 00000039			RESET_STATUS_ONLY	0000045C R 03
IOS_NOP	= 00000000			RETREG	00000389 R 03
IOS_PACKACK	= 00000008			RETRYERR	00000252 R 03
IOS_READBLK	= 00000021			RETURNED_STATUS	= 00000004
IOS_READPBLK	= 0000000C			SEEK	0000022A R 03
IOS_READVBLK	= 00000031			SIZ...	= 00000008
IOS_SEEK	= 00000002			SPECOND	00000414 R 03
IOS_SENSECHAR	= 0000001B			SSS_CTRLERR	= 00000054
IOS_SENSEMODE	= 00000027			SSS_DRVERR	= 0000008C
IOS_SETCHAR	= 0000001A			SSS_IVBUFLEN	= 0000034C
IOS_SETMODE	= 00000023			SSS_MEDOFL	= 000001A4
IOS_UNLOAD	= 00000001			SSS_NORMAL	= 00000001
IOS_VIRTUAL	= 0000003F			SSS_PARITY	= 000001F4
IOS_WRITEBLK	= 00000020			SSS_TIMEOUT	= 0000022C
IOS_WRITEPBLK	= 0000000B			SSS_VOLINV	= 00000254
IOS_WRITEVBLK	= 00000030			SSS_WRITLCK	= 0000025C
IOCSDIAGBUFILL	*****	X	03	STATUS_RESET	= 00000002
IOCSMNTVER	*****	X	03	STXCS_HIE	= 00000040
IOCSMOVFRUSER	*****	X	03	STXCS_S_ADDRS	= 00000010
IOCSMOVFRUSER2	*****	X	03	STXCS_S_STS	= 00000008
IOCSMOVTOUSER	*****	X	03	STXCS_V_ADDRS	= 00000008
IOCSMOVTOUSER2	*****	X	03	STXCS_V_RDY	= 00000007
IOCSPRELCHAN	*****	X	03	STXCS_V_STS	= 00000018
IOCSPREQCOM	*****	X	03	TRANS_ABORTED	= 00000003
IOCSPREQCHANL	*****	X	03	TRANS_COMPLETE	= 00000001
IOCSPRETURN	*****	X	03	TRANS_CONTINUE	= 00000002
IOCSPWF1KPCH	*****	X	03	UCBSB_CV_STATE	000000DC
IRPSL_MEDIA	= 00000038			UCBSB_CV_STS	000000DD
IRPSL_SVAPTE	= 0000002C			UCBSB_DEVCLASS	= 00000040
IRPSS_FCODE	= 00000006			UCBSB_DEVTTYPE	= 00000041
IRPSV_DIAGBUF	= 00000007			UCBSB_DIPL	= 0000005E
IRPSV_FCODE	= 00000000			UCBSB_ERTCNT	= 00000080
IRPSV_PHYSIO	= 00000008			UCBSB_ERTMAX	= 00000081

UCBSB_FEX	=	00000092
UCBSB_IPL	=	00000008
UCBSB_SECTORS	=	00000044
UCBSB_TRACKS	=	00000045
UCBSK_CV_LEN	=	000000FC
UCBSK_LCC_DISK_LENGTH	=	000000CC
UCBSL_CRB	=	C0000024
UCBSL_CV_ABPC	=	000000F8
UCBSL_CV_BUFWIN	=	000000E8
UCBSL_CV_CS	=	000000CC
UCBSL_CV_IBUF	=	000000E0
UCBSL_CV_LBN	=	000000F4
UCBSL_CV_MP	=	000000D0
UCBSL_CV_MVRTN	=	000000E4
UCBSL_DEVCHAR	=	00000038
UCBSL_DPC	=	0000009C
UCBSL_FPC	=	0000000C
UCBSL_IRP	=	00000058
UCBSL_MAXBLOCK	=	000000B0
UCBSL_MEDIA	=	000000BC
UCBSL_MEDIA_ID	=	0000008C
UCBSL_SVAPTE	=	00000078
UCBSM_DIAGBUF	=	00000002
UCBSM_INT	=	00000002
UCBSM_ONLINE	=	00000010
UCBSM_POWER	=	00000020
UCBSM_TIMOUT	=	00000040
UCBSM_VALID	=	00000800
UCBSQ_CV_BDAT	=	000000EC
UCBSQ_CV_CSMP	=	000000D4
UCBSV_INT	=	00000001
UCBSV_POWER	=	00000005
UCBSV_VALID	=	00000008
UCBSW_BCNT	=	0000007E
UCBSW_BCR	=	000000C0
UCBSW_CV_BBC	=	000000DE
UCBSW_CVINDERS	=	00000046
UCBSW_DEVBUFFSIZ	=	00000042
UCBSW_DEVSTS	=	00000068
UCBSW_FUNC	=	0000009A
UCBSW_STS	=	00000064
UNLOAD	=	0000023C R 03
VECSL_IDB	=	00000008
VECSL_INITIAL	=	0000000C
VECSL_UNITINIT	=	00000018
WRITECHECK	=	0000022A R 03
WRITEDATA	=	00000249 R 03
WRITE_BLOCK	=	00000005
XFER	=	00000324 R 03

```
+-----+
! Psect synopsis !
+-----+
```

## PSECT name

	Allocation	PSECT No.	Attributes
ABS	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	000000FC ( 252.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
\$_\$S105_PROLOGUE	0000007C ( 124.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$_\$S115_DRIVER	0000079D ( 1949.)	03 ( 3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

	Allocation	PSECT No.	Attributes
ABS	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	000000FC ( 252.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
\$_\$S105_PROLOGUE	0000007C ( 124.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
\$_\$S115_DRIVER	0000079D ( 1949.)	03 ( 3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

```
+-----+
! Performance indicators !
+-----+
```

## Phase

Phase	Page faults	CPU Time	Elapsed Time
Initialization	34	00:00:00.04	00:00:01.74
Command processing	118	00:00:00.42	00:00:05.63
Pass 1	549	00:00:17.57	00:01:13.91
Symbol table sort	0	00:00:02.41	00:00:09.60
Pass 2	238	00:00:03.73	00:00:12.75
Symbol table output	33	00:00:00.20	00:00:00.33
Psect synopsis output	3	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	977	00:00:24.40	00:01:44.13

The working set limit was 2100 pages.

140986 bytes (276 pages) of virtual memory were used to buffer the intermediate code.

There were 120 pages of symbol table space allocated to hold 2192 non-local and 53 local symbols.

1303 source lines were read in Pass 1, producing 21 object records in Pass 2.

45 pages of virtual memory were used to define 42 macros.

```
+-----+
! Macro library statistics !
+-----+
```

## Macro library name

Macro library name	Macros defined
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	28
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	10
TOTALS (all libraries)	38

## Macros defined

Macro library name	Macros defined
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	28
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	10
TOTALS (all libraries)	38

2392 GETS were required to define 38 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:CVDRIVER/OBJ=OBJ\$:CVDRIVER MSRC\$:CVDRIVER/UPDATE=(ENH\$:(CVDRIVER)+EXECMLS/LIB

0108 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

